



# CHAPTER 1

Understanding Networks and their Building Blocks

U TUN MIN OO (BE-IT)

## Chapter 1 – *Understanding Networks and their Building Blocks*

### 1-1 Introduction to Networks

Before you learn Cisco Internet working, it is important to understand what a network is and the importance of networks themselves. Simply put, a network is a collection of interconnected devices (such as computers, printers, etc.). To understand the importance of networks, let us look at how things worked before networks were created. For this, consider a large multinational company that sells food products in a time when networks did not exist.

Let us call this company ABC Inc. Imagine the amount of information such as sales, inventory, etc. required by the management of the company to make everyday decisions. To get this information they will need to call their local offices. Their local offices will need to mail (postal!) or fax printed reports or even send media (floppies!) through the postal service. By the time the mail is received, the data is already days old. Even if reports are faxed, it will be a cumbersome task to consolidate all reports. This task also increases chance of human error since large numbers of reports are manually collated. This is just one part of the equation. You also need to consider the information required by the local offices. They also need various data from the head office and other offices around the world.

Now consider the same company, but in the present time with all their offices interconnected. They would use a single application around the world that takes advantage of their global network. The data from all offices would be instantly stored at the central location and with a single click, the management team can see data from around the world in any format they like. This data would also be real-time. This means that they see it as it's happening. Since the data is centralized, any office location can see data pertaining to any location.

As you can see, the cost, time and effort involved in transferring data was much higher without networks. So networks decrease cost, time, and effort and thereby increase productivity. They also help in resource optimization by helping to share resources. A simple example of resource sharing is a printer in a typical office. Without networks, each computer would require a dedicated printer. However with a network, the printer can be shared between many different computers.

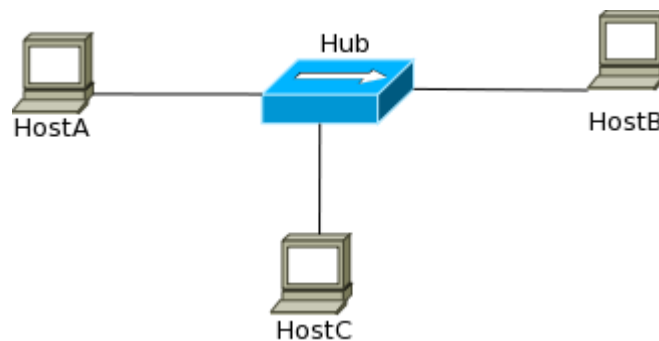
Now that you know how beneficial networks are, it's time to look at how networks work. Figure 1-1 shows the most basic form of a network. This figure shows two hosts (end-user devices such as computers are commonly called hosts in networking terms) directly connected to each other using a networking cable. Today every host has a **Network Interface Card (NIC)** that is used to connect it to a network.



**Figure 1-1** *Most basic form of Network*

One end of the network cable connects to the NIC on a host and the other connects to the network. In this case, the cable directly connects to another host. At this stage do not worry about network cables and how the hosts communicate across the network. This will be covered in detail later in the chapter. At this stage it is important to understand how hosts connect to a network.

In Figure 1-1, the hosts are “networked” and can share information. This network is effective, but not scalable. If you have more than 2 hosts to this “network”, it will not work without a separate NIC card for each connection and that is not scalable or realistic. For more than 2 hosts to be networked, you require a network device such as a **hub**. Figure 1-2 shows three hosts connected to a hub.



**Figure 1-2** *Network with a Hub*

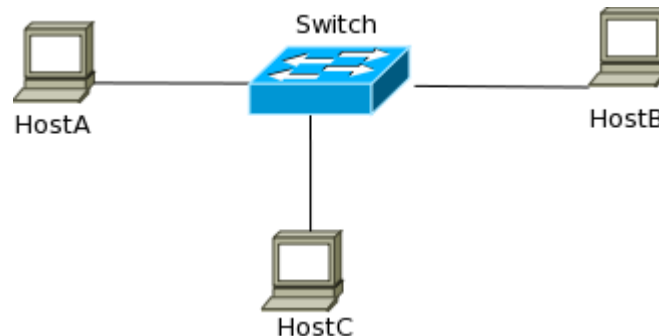
A hub is a network device that repeats information received from a host to all other connects hosts. In Figure 1-2 the hub will relay any information received from HostA to HostB and HostC. This means that all the three hosts can communicate with each other. Communication between hosts can be classified into three types:

- **Unicast** – Communication from one host to another host only.
- **Broadcast** – Communication from one host to all the hosts in the network.
- **Multicast** – Communication from one host to few hosts only.

When a hub is used to network hosts, there are two problems that arise:

1. A hub repeats information received from one host to all the other hosts. To understand this, consider HostA in Figure 1-2 sending a unicast message to HostB. When the hub receives this message; it will relay the message to both HostB and HostC. Even though the message was a unicast intended only for HostB, HostC also receives it. It is up to HostC to read the message and discard it after seeing that the message was not intended for it.
2. A hub creates a shared network medium where only a single host can send packets at a time. If another host attempts to send packets at the same time, a collision will occur. Then each device will need to resend their packets and hope not to have a collision again. This shared network medium is called a single **collision domain**. Imagine the impact of having a single collision domain where 50 or 100 hosts are connected to hubs that are interconnected and they are all trying to send data. That is just a recipe for many collisions and an inefficient network.

The problems associated with hubs can cause severe degradation of a network. To overcome these, **switches** are used instead of hubs. Like hubs, switches are used to connect hosts in a network but switches break up collision domain by providing a single collision domain for every port. This means that every host (one host connects to one port on the switch) gets its own collision domain thereby eliminating the collisions in the network. With switches, each host can transmit data anytime. Switches simply “switch” the data from one port to another in the switched network. Also, unlike hubs, switches do not flood every packet out all ports. They switch a unicast packet to the port where the destination host resides. They only flood out a broadcast packet. Figure 1-3 shows a switched network.



**Figure 1-3 A switched network**

Remember that each host in Figure 1-3 is in its own collision domain and if HostA sends a packet to HostC, HostB will not receive it.

Figure 1-4 and 1-5 show two networks. See if you can figure out how many collision domains exist in them.

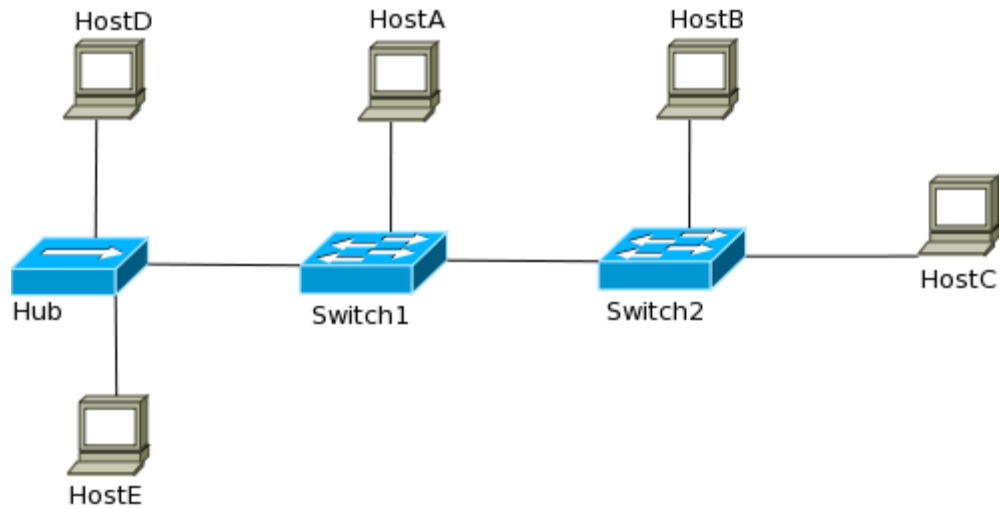


Figure 1-4 Collision Domains – 1

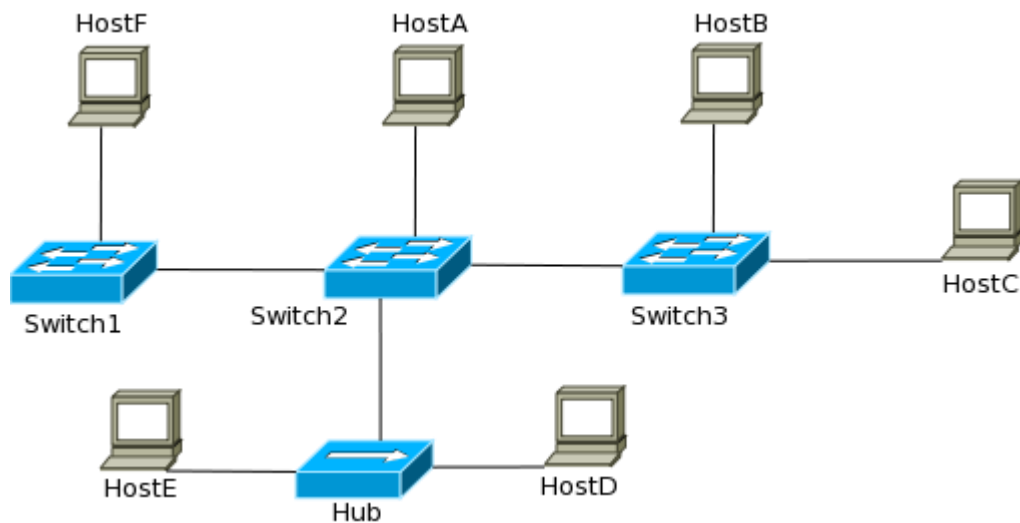
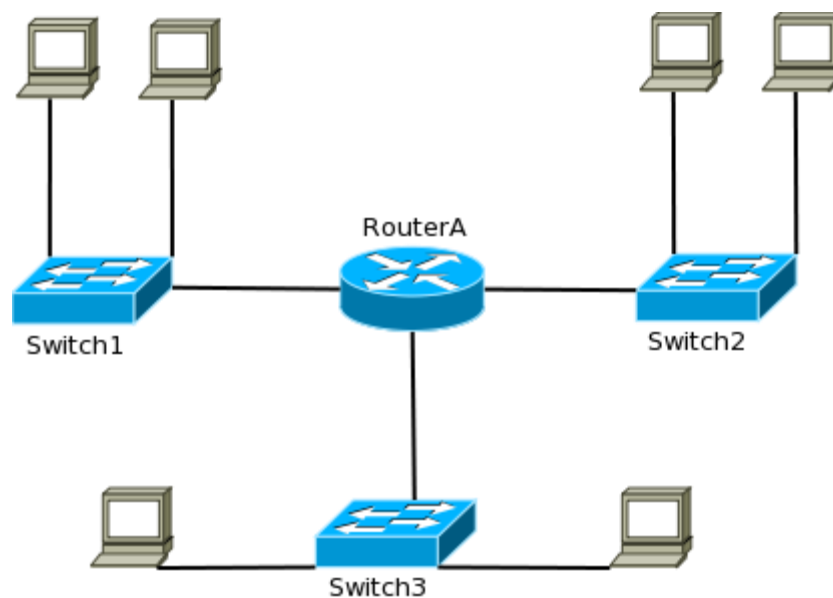


Figure 1-5 Collision Domains – 2

If you answered 5 for Figure 1-4, then you are absolutely correct since each port of the Switches represent a single collision domain. If you answered more than 5 then you need to remember that a hub does not break collision domains. Similarly, Figure 1-5 has 7 collision domains.

Now that you know how a switch works and improves a network, consider the one problem associated with a switched network. Earlier, you learned that hubs flood out all packets, even the unicast ones. A switch does not flood out unicast packets but it does flood out a broadcast packet. All hosts connected to a switched network are said to be in the same **broadcast domain**. All hosts connected to it will receive any broadcast sent out in this domain. While broadcasts are useful and essential for network operations, in a large switched network too many broadcasts will slow down the network. To remedy this situation, networks are broken into smaller sizes and these separate networks are interconnected using **routers**. Routers do not allow broadcasts to be transmitted across different networks it interconnects and hence effectively breaks up a broadcast domain. Figure 1-6 shows three switched networks interconnected by a router.



**Figure 1-6 Router in an Internetwork**

In the network shown in Figure 1-6, broadcasts from hosts connected to Switch1 will not reach hosts connected to Switch2 or Switch3. This is because the router will drop the broadcast on its receiving interface.

In addition to breaking up broadcast domains, routers also perform the following four essential functions in your network:

- **Packet Switching** – At the barest minimum, routers are like switches because they essentially switch packets between networks.
- **Communication between Networks** – As shown in Figure 1-6, routers allow communication between networks connected to it.
- **Path Selection** – Routers can talk to each other to learn about all the networks connected to various routers and then select the best path to reach a network. This function is discussed in detail later in the book.
- **Packet Filtering** – Routers can drop or forward packets based on certain criteria like their source and destination. This is also discussed in detail later in the book.
- ***Now that you know what a network is and what various network devices do, it's time to learn about various network types followed by networking models.***

◆◆◆◆ GO TO 1-2◆◆◆◆

## **1-2 Networking Types**

As you know a network is a collection of devices connected together. Networks are further classified into various types depending on their size, expanse, security, purpose and many other parameters. While covering all these classifications is beyond the scope of the CCNA exam, there are two important network classifications that you need to know about for the exam. In fact a large part of the CCNA exam revolves around these two types of networks:

- **Local Area Network (LAN)** – This is a term used to describe a network covering a limited geographical area such as a floor, building or a campus. LAN usually has a high data-transfer rate. The Ethernet standard is the most commonly used technology in LANs. Ethernet is so common that it is almost synonymous with LAN today. As of late, wireless technology is also becoming increasingly common for a local LAN. Both these standards are covered in depth further in the book.
- **Wide Area Network (WAN)** – This is a term used to describe a network covering a large geographical area such as a multiple cities, a country or even across the world. They are used to connect LANs across the area they cover. A typical example would be the LANs at various offices of a company connected by WAN. Various technology standards used in WAN will be covered later in the book.

## Internetworking Models

As the importance of computers grew, vendors recognized the need for networking them. They created various protocols whose specifications were not made public. Hence each vendor had different ways of networking computers and these ways were not compatible to each other. This means that computers of one vendor could not be networked with another vendor's computers. Slowly these specifications were made public and some inter-vendor compatibility was created but this still represented too many complications. In 1977 the International Organization for Standardization (ISO) started working on an open standard networking model that all vendors would support to promote inter-operability. This standard was published in 1984 and was known as the **Open Systems Interconnection (OSI)**. During the same time period (1973 to 1985) another effort by the Defense Advanced Research Projects Agency (DAPRA) was underway to create an open standard network model. This network model came to be known as the **TCP/IP Model**. By 1985, the TCP/IP model started gaining more prominence and support from vendors and eventually replaced the OSI model.

This section starts by discussing the OSI Reference model in some depth before moving into a deep discussion on the TCP/IP model and its protocols.

◆◆◆◆◆ GO TO 1-3 ◆◆◆◆◆

### **1-3 OSI Reference Model**

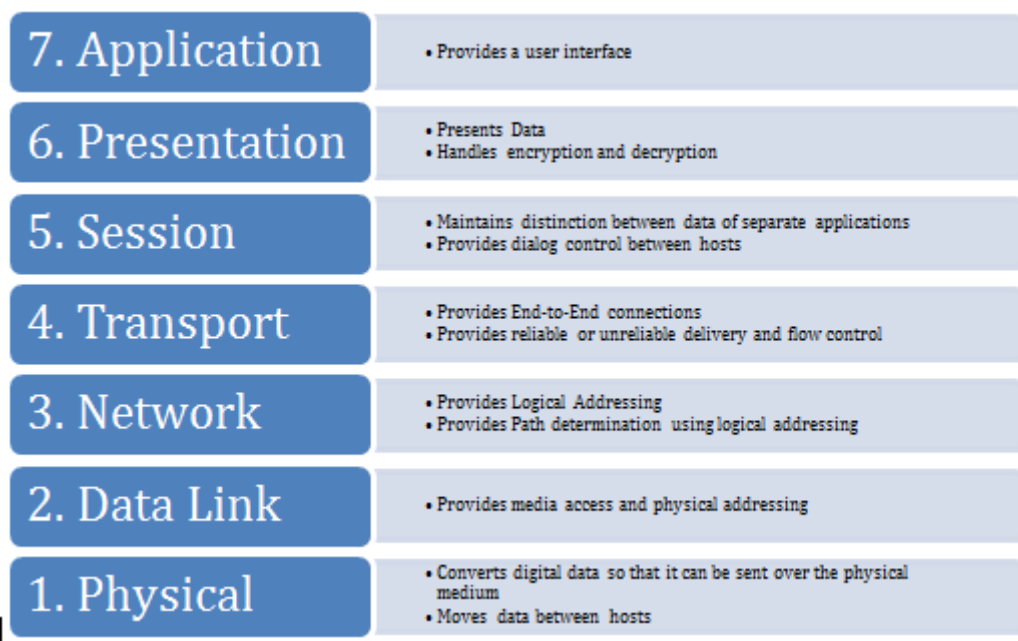
As discussed earlier, the OSI model was created to promote communication between devices of various vendors. It also promotes communication between disparate hosts such as hosts using different operating platforms (Windows, OSX, Linux, etc.). Remember that you are very unlikely to ever work on a system that uses protocols conforming to the OSI reference model. But it is essentially to know the model and its terminology because other models such as the TCP/IP model are often compared to the OSI reference model. Hence the discussion on this model will be limited compared to the discussion on the TCP/IP model.



The OSI reference model, like most other network models, divides the functions, protocols, and devices of a network into various layers. The layered approach provides many benefits, some of which are:

- Communication is divided into smaller and simpler components. This makes designing, developing and troubleshooting easier.
- Since it is a layered approach, the vendors write to a common input and output specification per layer. The guts of their products functions in between the input and output code of that layer.
- Changes in one layer do not affect other layers. Hence development in one layer is not bound by limitations of other layers. For example, wireless technologies are new but old applications run seamless over them without any changes.
- It is easier to standardize functions when they are divided into smaller parts like this.
- It allows various types of hardware and software, both new and old to communicate with each other seamlessly.

The OSI reference model has seven such layers that can be divided into two groups. The upper layers (Layers 7, 6 and 5) define how applications interact with the host interface, with each other, and the user. The lower four layers (Layers 4, 3, 2 and 1) define how data is transmitted between hosts in a network. Figure 1-7 shows the seven layers and a summary of their functions.



**Figure 1-7 Seven Layers of OSI Reference Model**

The sections below discuss each layer in detail.

### Application Layer

The Application Layer provides the interface between the software application on a system and the network. Remember that this layer does not include the application itself, but provides services that an application requires. One of the easiest ways to understand this layer's function is to look at how a Web Browser such as Internet Explorer or Firefox works. IE or FF is the application. When it needs to fetch a webpage, it uses the **HTTP** protocol to send the request and receive the page contents. This protocol resides at the application layer and can be used by an application such as IE or FF to get webpages from web servers across the network. On the other side, the web server application such as Apache or IIS interacts with the HTTP protocol on the Application layer to receive the HTTP request and send the response back.

### Presentation Layer

As the name suggests, this layer presents data to the Application layer. The Presentation Layer is responsible for data translation and encoding. It will take the data from the Application layer and translate it into a generic format for transfer across the network. At the receiving end the Presentation layer takes in generically formatted data and translates into the format recognized by the Application layer. An example of this is an **EBCDIC** to **ASCII** translation. The OSI model has protocol standards that define how data should be formatted. This layer is also involved in data compression, decompression, encryption, and decryption.

### Session Layer

In a host, different applications or even different instances of the same application might request data from across the network. It is the Sessions layer's responsibility to keep the data from each session separate. It is responsible for setting up, managing and tearing down sessions. It also provides dialog control and coordinates communication between the systems.

### Transport Layer

Where the upper layers are related to applications and data within the host, the transport layer is concerned with the actual end-to-end transfer of the data across the network. This layer establishes a logical connection between the two communicating hosts and provides reliable or unreliable data delivery and can provide flow control and error recovery. Although not developed under the OSI Reference Model and not strictly conforming to the OSI definition of the Transport Layer, typical examples of Layer 4 are the **Transmission Control Protocol (TCP)** and **User Datagram Protocol (UDP)**. These protocols will be discussed in great detail later in this chapter.

### Network Layer

To best understand what the Network layer does, consider what happens when you write a letter and use the postal service to send the letter. You put the letter in an envelope and write the destination address as well as your own address so that an undelivered letter can be returned back to you. In network terms, this address is called a logical address and is unique in the network. Each host has a logical address. When the post office receives this letter, it has to ascertain the best path for this letter to reach the destination. Similarly in a network, a router needs to determine the best path to a destination address. This is called path determination. Finally the post office sends the letter out the best path and it moves from post office to post office before finally being delivered to the destination address. Similarly data is moved across network mainly by routers before being finally delivered to the destination.

All these three functions – logical addressing, path determination and forwarding – are done at the Network Layer. Two types of protocols are used for these functions – **routed protocols** are used for logical addressing and forwarding while **routing protocols** are used for path determinations. There are many routed protocols and routing protocols available. Some of the common ones are discussed in great detail later the book. Routers function at this layer. Remember that routers only care about the destination network. They do not care about the destination host itself. The task of delivery to the destination host lies on the Data Link Layer.

### Data Link Layer

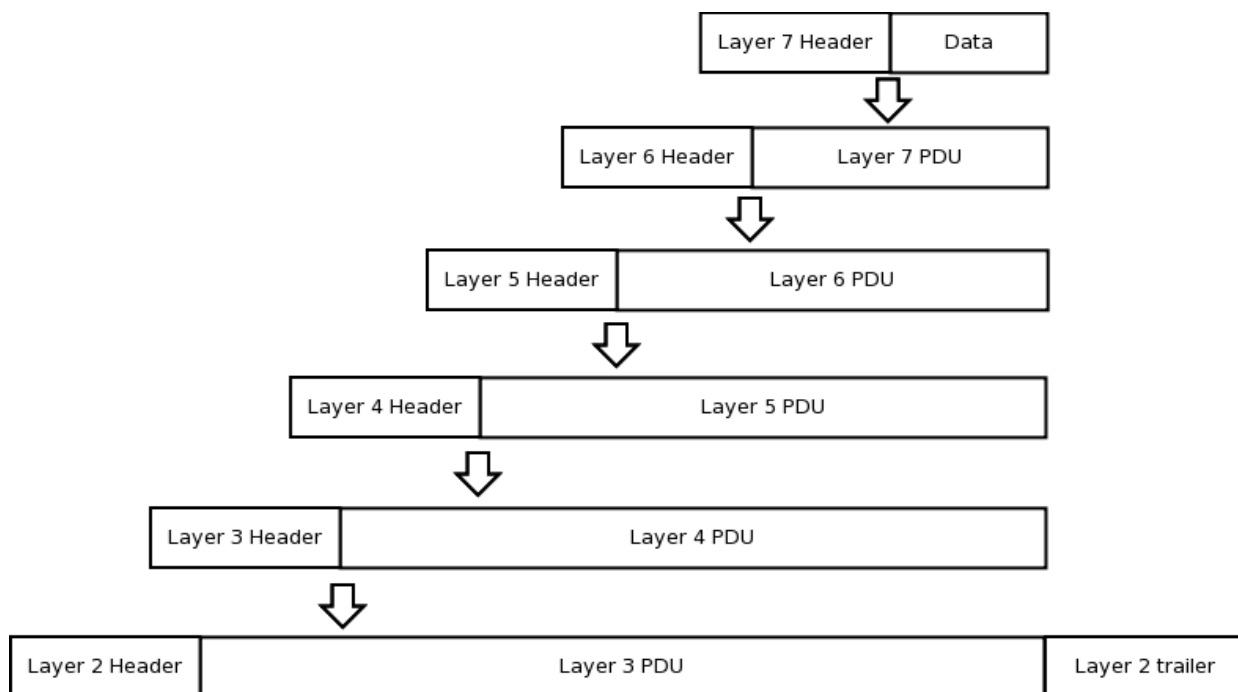
While the Network layer deals with data moving across networks using logical addresses, Data Link layer deals with data moving within a local network using physical addresses. Each host has a logical address and a physical address. The physical address is only locally significant and is not used beyond the network boundaries (across a router). This layer also defines protocols that are used to send and receive data across the media. You will remember from earlier in the chapter that only a single host can send data at a time in a collision domain or else packets will collide and cause a host to back off for sometime. The Data Link layer determines when the media is ready for the host to send the data and also detects collisions and other errors in received data. Switches function at this layer.

### Physical Layer

This layer deals with the physical transmission medium itself. It activates, maintains and deactivates the physical link between systems (host and switch for example). This is where the connectors, pin-outs, cables, electrical currents etc. are defined. Essentially this layer puts the data on the physical media as bits and receives it in the same way. Hubs work at this layer.

## Data Encapsulation

In the previous sections you learned about various layers of the OSI reference model. Each layer has its distinct function and it interacts with the corresponding layer at the remote end. For example, the transport layer at the source will interact with the transport layer of the destination. For this interaction, each layer adds a header in front of the data from the previous layer. This header contains control information related to the protocol being used at that layer. This process is called **encapsulation**. This header and the data being sent from one layer to the next lower layer is called a **Protocol Data Unit (PDU)**. Figure 1-8 shows how data gets encapsulated as it travels from layer 7 down to layer 1.



**Figure 1-8 Encapsulation in OSI Reference Model**

As shown in Figure 1-8, The Application layer adds its protocol dependent header to the data and creates the Layer 7 PDU which is then passed down to the Presentation Layer. This layer then adds its header to the Layer 7 PDU to create the Layer 6 PDU and sends it down to the Session layer. This goes on till Layer 2 receives the Layer 3 PDU. Layer 2 adds a header and a trailer to the Layer 3 PDU to create the Layer 2 PDU that is then sent to Layer 1 for transmission.

At the receiving end, Layer 1 takes the data off the wire and sends it to Layer 2. Here the Layer 2 header and trailer are examined and removed. The resulting Layer 3 PDU is sent to Layer 3. Layer 3 in turn examines the header in the PDU and removes it. The resulting Layer 4 PDU is sent to Layer 4. Similarly, each layer removes the header added by the corresponding layer at the source before sending the data to the upper layer. Finally the Application layer removes the Layer 7 header and sends the data to the application. This process of examining, processing and removing the header is known as **decapsulation**.

◆◆◆◆◆ GO TO 1-4 ◆◆◆◆◆

### **1-4 TCP/IP Model**

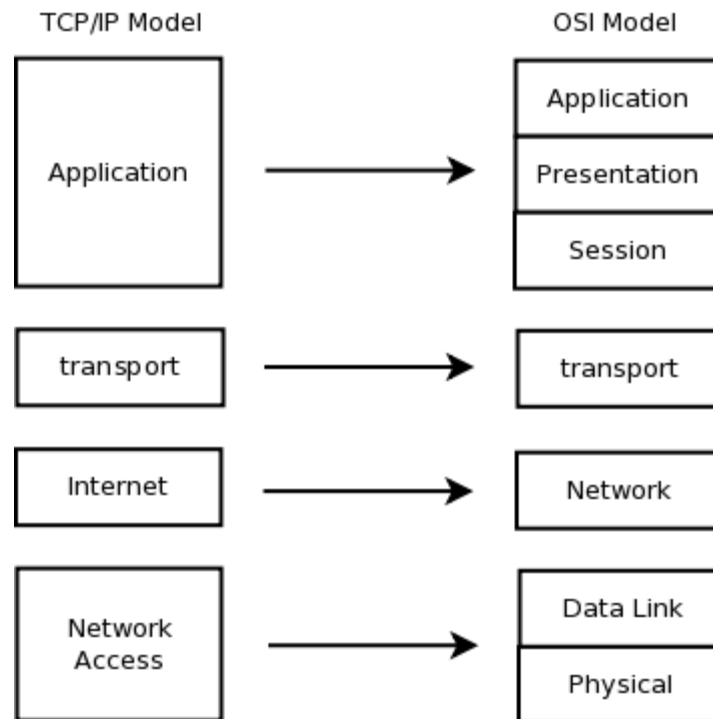
As mentioned earlier, the OSI reference model and the TCP/IP model are two open standard networking models that are very similar. However, the latter has found more acceptance today and the TCP/IP protocol suite is more commonly used. Just like the OSI reference model, the TCP/IP model takes a layered approach. In this section we will look at all the layers of the TCP/IP model and various protocols used in those layers.

The TCP/IP model is a condensed version of the OSI reference model consisting of the following 4 layers:

- Application Layer
- Transport Layer
- Internet Layer
- Network Access Layer

The functions of these four layers are comparable to the functions of the seven layers of the OSI model. Figure 1-9 shows the comparison between the layers of the two models.

The following sections discuss each of the four layers and protocols in those layers in detail.



**Figure 1-9** *Comparison between TCP/IP and OSI models*

### Application Layer

The Application Layer of the TCP/IP Model consists of various protocols that perform all the functions of the OSI model's Application, Presentation and Session layers. This includes interaction with the application, data translation and encoding, dialogue control and communication coordination between systems.

The following are few of the most common Application Layer protocols used today:

**Telnet** – Telnet is a terminal emulation protocol used to access the resources of a remote host. A host, called the Telnet server, runs a telnet server application (or daemon in Unix terms) that receives a connection from a remote host called the Telnet client. This connection is presented to the operating system of the telnet server as though it is a terminal connection connected directly (using keyboard and mouse). It is a text-based connection and usually provides access to the command line interface of the host. Remember that the application used by the client is usually named telnet also in most operating systems. You should not confuse the *telnet* application with the Telnet protocol.

**HTTP** – The Hypertext Transfer Protocol is foundation of the World Wide Web. It is used to transfer Webpages and such resources from the Web Server or HTTP server to the Web Client or the HTTP client. When you use a web browser such as Internet Explorer or Firefox, you are using a web client. It uses HTTP to transfer web pages that you request from the remote servers.

**FTP** – File Transfer Protocol is a protocol used for transferring files between two hosts. Just like telnet and HTTP, one host runs the FTP server application (or daemon) and is called the FTP server while the FTP client runs the FTP client application. A client connecting to the FTP server may be required to authenticate before being given access to the file structure. Once authenticated, the client can view directory listings, get and send files, and perform some other file related functions. Just like telnet, the FTP client application available in most operating systems is called *ftp*. So the protocol and the application should not be confused.

**SMTP** – Simple Mail Transfer Protocol is used to send e-mails. When you configure an email client to send e-mails you are using SMTP. The mail client acts as a SMTP client here. SMTP is also used between two mails servers to send and receive emails. However the end client does not receive emails using SMTP. The end clients use the **POP3** protocol to do that.

**TFTP** – Trivial File Transfer Protocol is a stripped down version of FTP. Where FTP allows a user to see a directory listing and perform some directory related functions, TFTP only allows sending and receiving of files. It is a small and fast protocol, but it does not support authentication. Because of this inherent security risk, it is not widely used.

**DNS** – Every host in a network has a logical address called the **IP address** (discussed later in the chapter). These addresses are a bunch of numbers. When you go to a website such as [www.cisco.com](http://www.cisco.com) you are actually going to a host which has an IP address, but you do not have to remember the IP Address of every WebSite you visit. This is because Domain Name Service (DNS) helps map a name such as [www.cisco.com](http://www.cisco.com) to the IP address of the host where the site resides. This obviously makes it easier to find resources on a network. When you type in the address of a website in your browser, the system first sends out a DNS query to its DNS server to resolve the name to an IP address. Once the name is resolved, a HTTP session is established with the IP Address.

**DHCP** – As you know, every host requires a logical address such as an IP address to communicate in a network. The host gets this logical address either by manual configuration or by a protocol such as Dynamic Host Configuration Protocol (DHCP). Using DHCP, a host can be provided with an IP address automatically. To understand the importance of DHCP, imagine having to manage 5000 hosts in a network and assigning them IP address manually! Apart from the IP address, a host needs other information such as the address of the DNS server it needs to contact to resolve names, gateways, subnet masks, etc. DHCP can be used to provide all these information along with the IP address.

### Transport Layer

The protocols discussed above are few of the protocols available in the Application layer. There are many more protocols available. All of them take the user data and add a header and pass it down to the Transport layer to be sent across the network to the destination. The TCP/IP transport layer's function is same as the OSI layer's transport layer. It is concerned with end-to-end transportation of data and setups up a logical connection between the hosts.

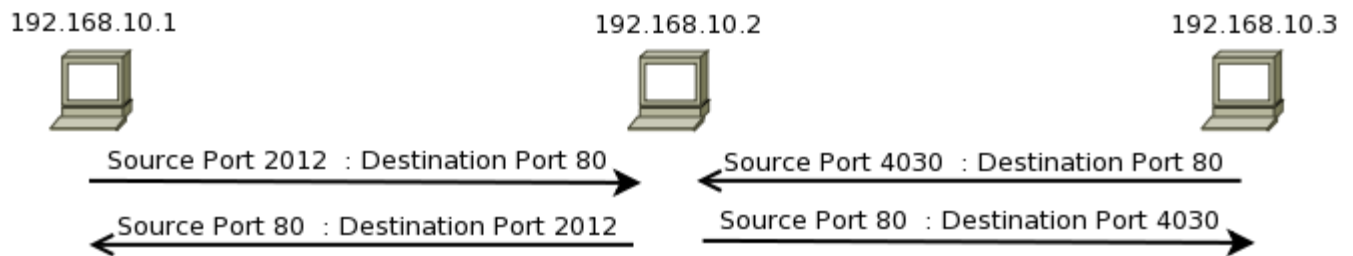
Two protocols available in this layer are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). TCP is a **connection oriented** and **reliable** protocol that uses **windowing** to control the flow and provides **ordered delivery** of the data in segments. On the other hand, UDP simply transfers the data without the bells and whistles. Though these two protocols are different in many ways, they perform the same function of transferring data and they use a concept called **port numbers** to do this. The following sections cover port numbers before looking into TCP and UDP in detail.

### Port Numbers

A host in a network may send traffic to or receive from multiple hosts at the same time. The system would have no way to know which data belongs to which application. TCP and UDP solve this problem by using port numbers in their header. Common application layer protocols have been assigned port numbers in the range of 1 to 1024. These ports are known as well-known ports. Applications implementing these protocols *listen* on these port numbers. TCP and UDP on the receiving host know which application to send the data to based on the port numbers received in the headers.

On the source host each TCP or UDP session is assigned a random port number above the range of 1024. So that returning traffic from the destination can be identified as belonging to the originating application. Combination of the IP address, Protocol (TCP or UDP) and the Port number forms a **socket** at both the receiving and sending hosts. Since each socket is unique, an application can send and receive data to and from multiple hosts. Figure 1-10 shows two hosts communicating using TCP. Notice that the hosts on the left and right are sending traffic to the host in the center and both of them are sending traffic destined to Port 80, but from different source ports. The host in the center is able to handle both the connections simultaneously because the combination of IP address, Port numbers and Protocols makes each connection different.





**Figure 1-10** Multiple Sessions using Port Numbers

Table 1-1 shows the transport layer protocol and port numbers used by different common application layer protocols.

Application Protocol	Transport Protocol	Port Number
HTTP	TCP	80
HTTPS	TCP	443
FTP (control)	TCP	21
FTP (data)	TCP	20
SSH	TCP	22
Telnet	TCP	23
DNS	TCP, UDP	53
SMTP	TCP	25
TFTP	UDP	69

**Table 1-1** Well-known Port Numbers



**Exam Alert:** It is important to remember the well-know port numbers and which application layer protocol they are assigned to as you will see this on your CCNA exam in a multiple choice question or an access-list question.

## Transport Control Protocol (TCP)

TCP is one of the original protocols designed in the TCP/IP suite and hence the name of the model. When the application layer needs to send large amount of data, it sends the data down to the transport layer for TCP or UDP to transport it across the network. TCP first sets up a virtual-circuit between the source and the destination in a process called **three-way handshake**. Then it breaks down the data into chunks called **segments**, adds a header to each segment and sends them to the Internet layer.

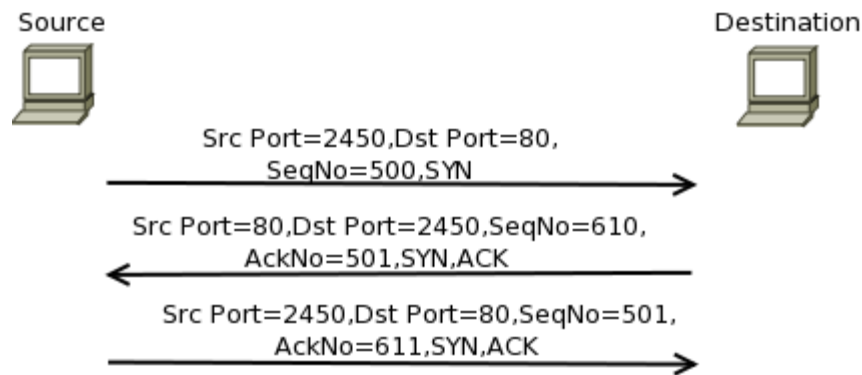
The TCP header is 20 to 24 bytes in size and the format is shown in Figure 1-11. It is not necessary to remember all fields or their size but most of the fields are discussed below.

Source Port (16 bits)			Destination Port (16 bits)		
Sequence Number (32 bits)					
Acknowledgement Number (32 bits)					
Header (4 bits)	Reserved (6 bits)	Code Bits (6 bits)	Window (16bits)		
Checksum (16bits)			Urgent (16bits)		
Options (0 to 32 bits)					

**Figure 1-11** *TCP header*

When the Application layer sends data to the transport layer, TCP sends the data across using the following sequence:

**Connection Establishment** – TCP uses a process called three-way handshake to establish a connection or virtual-circuit with the destination. The three-way handshake uses the **SYN** and **ACK** flags in the Code Bits section of the header. This process is necessary to initialize the sequence and acknowledgement number fields. These fields are important for TCP and will be discussed below.

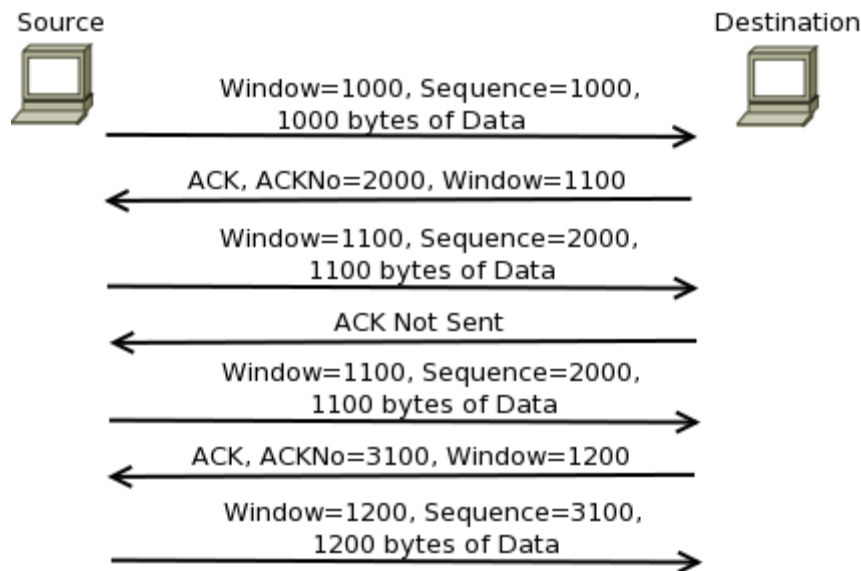


**Figure 1-12** *TCP three-way handshake*

As shown in Figure 1-12, the source starts the three-way handshake by sending a TCP header to the destination with the SYN flag set. The destination responds back with the SYN and ACK flag sent. Notice in the figure that destination uses the received sequence number plus 1 as the Acknowledgement number. This is because it is assumed that 1 byte of data was contained in the exchange. In the final step, the source responds back with only the ACK bit set. After this, the data flow can commence.

**Data Segmentation** – The size of data that can be sent across in a single Internet layer PDU is limited by the protocol used in that layer. This limit is called the **maximum transmission unit (MTU)**. The application layer may send data much larger than this limit; hence TCP has to break down the data into smaller chunks called segments. Each segment is limited to the MTU in size. Sequence numbers are used to identify each byte of data. The sequence number in each header signifies the byte number of the first byte in that segment.

**Flow Control** – The source starts sending data in groups of segments. The Window bit in the header determines the number of segments that can be sent at a time. This is done to avoid overwhelming the destination. At the start of the session the window is small but it increases over time. The destination host can also decrease the window to slow down the flow. Hence the window is called the **sliding window**. When the source has sent the number of segments allowed by the window, it cannot send any further segments till an acknowledgement is received from the destination. Figure 1-13 shows how the window increases during the session. Notice the Destination host increasing the Window from 1000 to 1100 and then to 1200 when it sends an ACK back to the source.



**Figure 1-13 TCP Sliding Window and Reliable delivery**

**Reliable Delivery with Error recovery** – When the destination receives the last segment in the agreed window, it has to send an acknowledgement to the source. It sets the ACK flag in the header and the acknowledgement number is set as the sequence number of the next byte expected. If the destination does not receive a segment, it does not send an acknowledgement back. This tells the source that some segments have been lost and it will retransmit the segments. Figure 1-13 shows how windowing and acknowledgement is used by TCP. Notice that when source does not receive acknowledgement for the segment with sequence number 2000, it retransmits the data. Once it receives the acknowledgement, it sends the next sequence according to the window size.

**Ordered Delivery** – TCP transmits data in the order it is received from the application layer and uses sequence number to mark the order. The data may be received at the destination in the wrong order due to network conditions. Thus TCP at the destination orders the data according to the sequence number before sending it to the application layer at its end. This order delivery is part of the benefit of TCP and one of the purposes of the Sequence Number.

**Connection Termination** – After all data has been transferred, the source initiates a four-way handshake to close the session. To close the session, the FIN and ACK flags are used.



**Exam Alert:** TCP is one of the most important protocols you will learn about while preparing for the CCNA exam. Understanding how TCP works is very important and you will more than likely see an ACK question on the exam!

## User Datagram Protocol (UDP)

The only thing common between TCP and UDP is that they use port numbers to transport traffic. Unlike TCP, UDP neither establishes a connection nor does it provide reliable delivery. UDP is **connectionless** and **unreliable** protocol that delivers data without overheads associated with TCP. The UDP header contains only four parameters (Source port, Destination Port, Length and Checksum) and is 8 bytes in size.

At this stage you might think that TCP is a better protocol than UDP since it is reliable. However you have to consider that networks now are far more stable than when these protocols were conceived. TCP has a higher overhead with a larger header and acknowledgements. The source also holds data till it receives acknowledgement. This creates a delay. Some applications, especially those that deal with voice and video, require fast transport and take care of the reliability themselves at the application layer. Hence in a lot of cases UDP is a better choice than TCP.

## Internet Layer

Once TCP and UDP have segmented the data and have added their headers, they send the segment down to the Network layer. The destination host may reside in a different network far from the host divided by multiple routers. It is the task of the Internet Layer to ensure that the segment is moved across the networks to the destination network.

The Internet layer of the TCP/IP model corresponds to the Network layer of the OSI reference model in function. It provides logical addressing, path determination and forwarding.

The **Internet Protocol (IP)** is the most common protocol that provides these services. Also working at this layer are routing protocols which help routers learn about different networks they can reach and the **Internet Control Message Protocol (ICMP)** that is used to send error messages across at this layer.

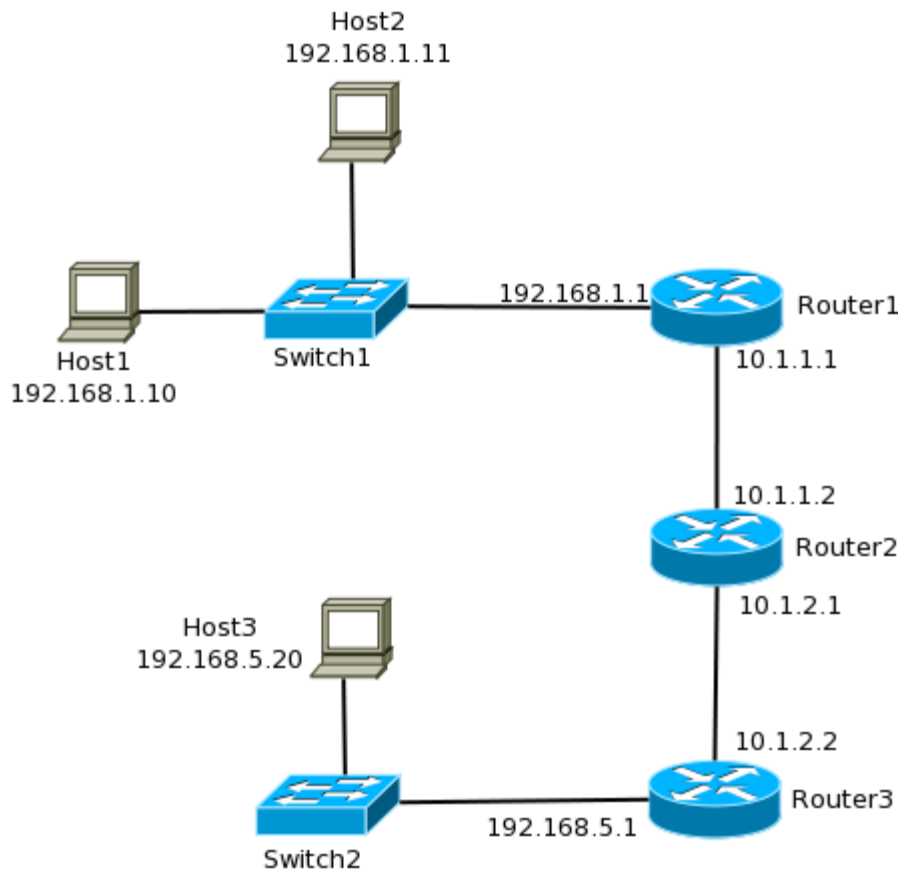
Almost half of the book is dedicated to IP and Routing protocols so they will be discussed in detail in later chapters, but the following sections discuss these protocols in brief.

## Internet Protocol (IP)

The Internet layer in the TCP/IP model is dominated by IP with other protocols supporting its purpose. Each host in a network and all interfaces of a router have a logical address called the IP address. All hosts in a network are grouped in a single IP address *range* similar to a street address with each host having a unique address from that range similar to a house or mailbox address. Each network has a different address range and routers that operate on layer 3 connect these different networks.

## TUN MIN OO {BE-IT} Routing & Switching 200-120

As IP receives segments from TCP or UDP, it adds a header with source IP address and destination IP address amongst other information. This PDU is called a **packet**. When a router receives a packet, it looks at the destination address in the header and forwards it towards the destination network. The packet may need to go through multiple routers before it reaches the destination network. Each router it has to go through is called a **hop**.



**Figure 1-14** *Packet flow in internetwork*

Consider the Internetwork shown in Figure 1-14 to understand the routing process better. When Host1 needs to send data to Host2, it does not get routed because the hosts are in the same network range. The Data Link layer takes care of this. Now consider Host1 sending data to Host3. Host1 will recognize that it needs to reach a host in another network and will forward the packet to Router1. Router1 checks the destination address and knows that the destination network is toward Router2 and hence forwards it to Router2. Similarly Router 2 forwards the packet to Router3. Router3 is directly connected to the destination network. Here the data link layer takes care of the delivery to the destination host. As you can see, the IP address fields in the IP header play a very important role in this process. In fact IP addresses are so important in a network that the next Chapter is entirely dedicated to it!

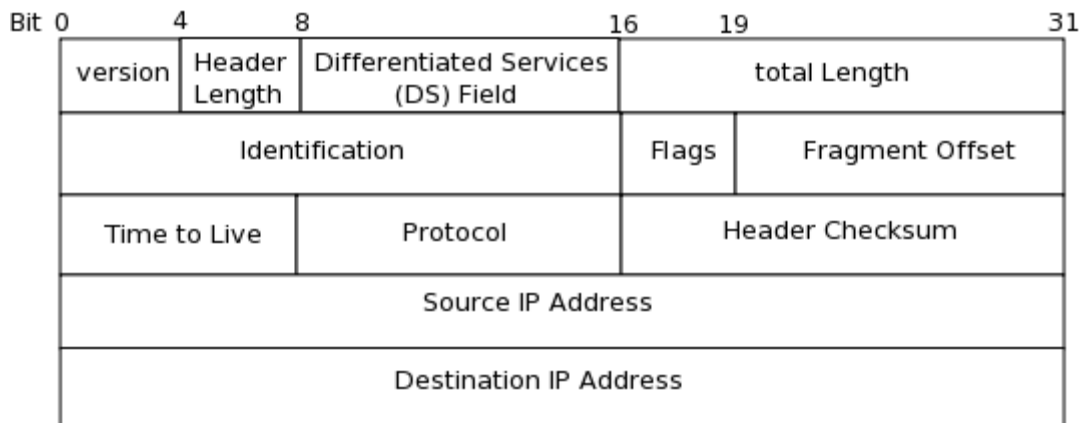


Figure 1-15 IPv4 Header

There are various versions of the Internet Protocol. Version 4 is the one used today and version 6 is slowly starting to replace it which is why its presence has increased on the CCNA Routing & Switching 200-120 exam compared to previous CCNA exam versions. Figure 1-15 shows the header structure of IPv4. The following fields make up the header:

**Version** – IP version number. For IPv4 this value is 4.

**Header Length** – This specifies the size of the header itself. The minimum size is 20 bytes. The figure does not show the rarely used options field that is of a variable length. Most IPv4 headers are 20 bytes in length.

**DS Field** – The differentiated Services field is used for marking packets. Different Quality-Of-Service (QoS) levels can be applied on different markings. For example, data belonging to voice and video protocols have no tolerance for delay. The DS field is used to mark packets carrying data belonging to these protocols so that they get priority treatment through the network. On the other hand, peer-to-peer traffic is considered a major problem and can be marked down to give in best effort treatment.

**Total Length** – This field specifies the size of the packet. This means the size of the header plus the size of the data.

**Identification** – When IP receives a segment from TCP or UDP; it may need to break the segment into chunks called **fragments** before sending it out to the network. Identification fields serves to identify the fragments that make up the original segment. Each fragment of a segment will have the same identification number.

**Flags** – Used for fragmentation process.

**Fragment Offset** – This field identifies the fragment number and is used by hosts to reassemble the fragments in the correct order.

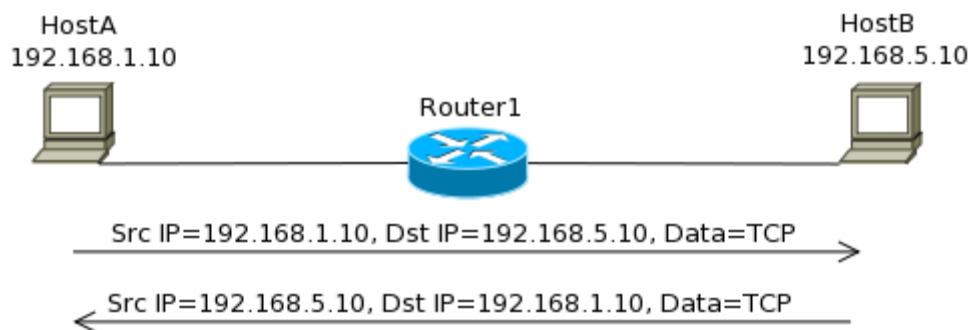
**Time to Live** – The Time to Live (TTL) value is set at the originating host. Each router that the packet passes through reduces the TTL by one. If the TTL reaches 0 before reaching the destination, the packet is dropped. This is done to prevent the packet from moving around the network endlessly.

**Protocol** – This field identifies the protocol to which the data it is carrying belongs. For example a value of 6 implies that the data contains a TCP segment while a value of 17 signifies a UDP segment. Apart from TCP and UDP there are many protocols whose data can be carried in an IP packet.

**Header Checksum** – This field is used to check for errors in the header. At each router and at the destination, a cyclic redundancy check performed on the header and the result should match the value stored in this field. If the value does not match, the packet is discarded.

**Source IP address** – This field stores the IP address of the source of the packet.

**Destination IP address** – This field stores the IP address of the destination of the packet.



**Figure 1-16** *Source and Destination IP address*

Figure 1-16 shows how Source and Destination IP address is used in an IP packet. Notice how the source and destination addresses changed during the exchange between HostA and HostB



## Routing Protocols

In Figure 1-14, Router1 knew that it needed to send the packet destined to Host3 toward Router2. Router2 in turn knew that the packet needed to go toward Router3. To make these decisions, the routers need to build their **routing table**. This is a table of all networks known by it and all the routers in the internetwork. The table also lists the next router towards the destination network. To build this table dynamically, routers use routing protocols. There are many routing protocols and their sole purpose is to ensure that routers know about all the networks and the best path to any network. Chapter 4 and Chapter 5 discuss the routing process and some routing protocols in detail.

## Internet Control Message Protocol (ICMP)

ICMP is essentially a management protocol and messaging service for IP. Whenever IP encounters an error, it sends ICMP data as an IP packet. Some of the reasons why an ICMP message can be generated are:

**Destination Network Unreachable** – If a packet cannot be routed to the network in which the destination address resides, the router will drop the packet and generate an ICMP message back to the source informing that the destination network is unreachable.

**Time Exceeded** – If the TTL of a packet expires (reduces to zero), the router will drop it and generate an ICMP message back to the source informing it that the time exceeded and the packet could not be delivered.

**Echo Reply** – ICMP can be used to check network connectivity. Popular utility called *Ping* is used to send *Echo Requests* to a destination. In reply to the request, the destination will send back an Echo reply back to the source. Successful receipt of Echo reply shows that the destination host is available and reachable from the source.

## Network Access Layer

The Network Access layer of the TCP/IP model corresponds with the Data Link and Physical layers of the OSI reference model. It defines the protocols and hardware required to connect a host to a physical network and to deliver data across it. Packets from the Internet layer are sent down the Network Access layer for delivery within the physical network. The destination can be another host in the network, itself, or a router for further forwarding. So the Internet layer has a view of the entire Internetwork whereas the Network Access layer is limited to the physical layer boundary that is often defined by a layer 3 device such as a router.

The Network Access layer consists of a large number of protocols. When the physical network is a LAN, **Ethernet** at its many variations are the most common protocols used. On the other hand when the physical network is a WAN, protocols such as the **Point-to-Point Protocol (PPP)** and **Frame Relay** are common. In this section we take a deep look at Ethernet and its variations. WAN protocols are covered in detail in Chapter 11.

Before we explore Ethernet remember that:

Network Access layer uses a physical address to identify hosts and to deliver data.

- The Network Access layer PDU is called a **frame**. It contains the IP packet as well as a protocol header and trailer from this layer.
- The Network Access layer header and trailer are only relevant in the physical network. When a router receives a frame, it strips of the header and trailer and adds a new header and trailer before sending it out the next physical network towards the destination.

♦♦♦♦♦ *GO TO 1-5* ♦♦♦♦♦

### **1-5 Ethernet Technologies and Cabling**

Ethernet is the term used for a family of standards that define the Network Access layer of the most common type of LAN used today. The various standards differ in terms of speeds supported, cable types and the length of cables. **The Institute of Electrical and Electronics Engineers (IEEE)** is responsible for defining the various standards since it took over the process in 1980.

To make it easier to understand Ethernet, its functions will be discussed in terms of the OSI reference models' Data Link and Physical layers. (Remember that Network Access Layer is a combination of these two layers).

IEEE defines various standards at the physical layer while it divides the Data Link functions into the following two sublayers:

- The 802.3 **Media Access Control (MAC)** sublayer
- The 802.2 **Logical Link Control (LLC)** sublayer

Even though various physical layer standards are different and require changes at the layer, each of them use the same 802.3 header and the 802.2 LLC sublayer.

The following sections look at the collision detection mechanism used by Ethernet and how Ethernet functions at both the layers.

### Collision Detection in Ethernet

Ethernet is a **contention media access** method that allows all hosts in a network to share the available bandwidth. This means that multiple hosts try to use the media to transfer traffic. If multiple hosts send traffic at the same time, a collision can occur resulting in loss of the frames that collided. Ethernet cannot prevent such collision but it can detect them and take corrective actions to resolve. It uses the **Carrier Sense Multiple Access with Collision Detection (CSMA/CD)** protocol to do so. This is how CSMA/CD works:

1. Hosts looking to transmit a frame listen until Ethernet is not busy.
2. When Ethernet is not busy, hosts start sending the frame.
3. The source listens to make sure no collision occurred.
4. If a collision occurs, the source hosts send a jamming signal to notify all hosts of the collision.
5. Each source host randomizes a timer and waits that long before resending the frame that collided.

CSMA/CD works well but it does create some performance issues because:

1. Hosts must wait till the Ethernet media is not busy before sending frames. This means only one host can send frames at a time in a collision domain (such as in the case of a network connected to a hub). This also means that a host can either send or receive at one time. This logic is called **half-duplex**.
2. During a collision, no frame makes it across the network. Also, the offending hosts must wait a random time before they can start to resend the frames.

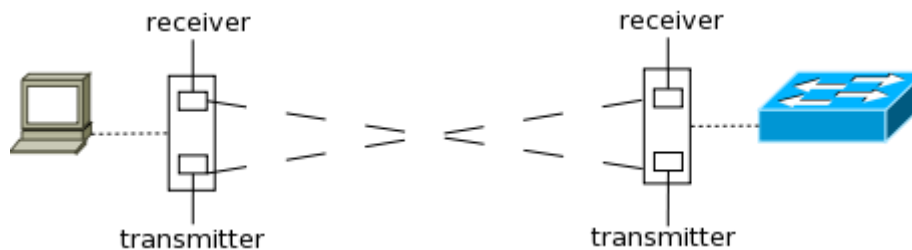
Many networks suffered this sort of performance degradation due to the use of hubs until switches became affordable. In fact, statistics showed that anything over 30 percent utilization caused performance degradation in Ethernet.

Remember that switches break collision domains by providing a dedicated port to each host. This means that hosts connected to a switch only need to wait if the switch is sending frames destined to the host itself.

### Half and Full Duplex Ethernet

In the previous section, you learned about the logic called **Half Duplex** in which a host can only send or receive at one time. In a hub-based network, hosts are connected in a half-duplex mode because they must be able to detect collisions.

When hosts are connected to a switch, they can operate at **Full duplex**. This means they can send and receive at the same time without worrying about collisions. This is possible because full duplex uses two pairs of wire instead of one pair. Using the two pairs, a point-to-point connection is created between the transmitter of the host to the receiver of the switch and vice versa. So the host sends and receives frames via different pairs of wires and hence need to listed to see if it send frames or not. You should note that CSMA/CD is disabled at both ends when full duplex is used.



**Figure 1-17 Full Duplex**

Apart from eliminating collisions, each device actually gets to use twice the bandwidth available because it now has same bandwidth on both pairs of wire and each pair is used separately for sending and receiving.

Figure 1-17 shows how the transmitter on the host's interface card is connected to the receiver on the switch interface while the receiver on the host interface is connected to the transmitter on the switch interface. Now traffic sent by the host and traffic sent to the host both have a dedicated path with equal bandwidth. If each path has a bandwidth of 100Mbps, the host gets 200Mbps of dedicated bandwidth to the switch. In case of half-duplex, there would have been only a single path of 100Mbps that would have been used for both receiving and sending traffic.

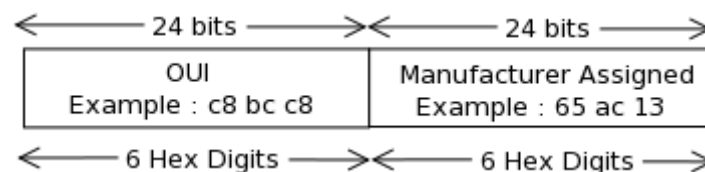
## Ethernet at the Data Link Layer

Ethernet at Data Link layer is responsible for addressing as well as framing the packets received from Network Layer and preparing them for the actual transmission.

### Ethernet Addressing

Ethernet Addressing identifies either a single device or a group of devices on a LAN and is called a **MAC address**. MAC address is 48 bits (6 bytes) long and is written in hexadecimal format. Cisco devices typically write it in a group of four hex digits separated by period while most operating systems write it in groups of two digits separated by a colon. For example, Cisco devices would write a MAC address as 5022.ab5b.63a9 while most operating systems would write it as 50:22:ab:5b:63:a9.

A **unicast** address identifies a single device. This address is used to identify the source and destination in a frame. Each LAN interface card has a globally unique MAC address. The IEEE defines the format and the assignment of addresses.



**Figure 1-18 48bit MAC address**

To keep addresses unique, each manufacturer of LAN cards is assigned a code called the **organizationally unique identifier (OUI)**. The first half of every MAC address is the OUI of the manufacturer. The manufacturer assigns the second half of the address while ensuring that the number is not used for any other card. The complete MAC address is then encoded into a ROM chip in the card. Figure 1-18 shows the composition of a MAC address.

MAC address can also identify a group of devices. These are called *group addresses*. IEEE defines the following two types of group addresses:

- **Broadcast Address** – This address has a value of FFFF.FFFF.FFFF and means that all devices in the network should process the frame.
- **Multicast Address** – Multicast addresses are used when a frame needs to go to a group of hosts in the network. When IP multicast packets need to travel over Ethernet a multicast address of 0100.5exx.xxxx is used where xx.xxxx can be any value.

## Ethernet Framing

When the Data Link layer receives a packet from the Network layer for transmission, it has to encapsulate the packet in frames. These frames are used to identify the source and destination device by the switch. It also tells the receiving host how to interpret the bits received by the physical layer.

<b>Preamble</b> 7	<b>SFD</b> 1	<b>Destination Address</b> - 6	<b>Source Address</b> - 6	<b>Length/ type</b> - 2	<b>Data</b> 46 - 1500	<b>FCS</b> 4
----------------------	-----------------	--------------------------------	---------------------------	-------------------------	--------------------------	-----------------

**Figure 1-19 IEEE Frame (1997)**

The framing used by Ethernet has changed few times over the year. Xerox defined the original frame. When IEEE took over Ethernet in early 1980s it defined a new frame. In 1997 IEEE finalized the Ethernet frame that took a few components from the Xerox definition and a few from IEEE's original frame. The finalized frame is shown in Figure 1-19. Table 1-2 lists the fields in the frame, their size and a brief description.

Field	Length in bytes	Description
<b>Preamble</b>	7	It is used for synchronization. It tells the received device where the header starts.
<b>SFD</b>	1	Start Frame Delimiter (SFD) tells the receiving device that the next byte is the destination address
<b>Destination Address</b>	6	Identifies the intended destination of the frame.
<b>Source Address</b>	6	Identifies the source of the frame.
<b>Length</b>	2	Contains the length of the data field of the frame. (This field can either be length or type but not both)
<b>Type</b>	2	Identifies the Network layer protocol whose data is contained in the frame. (This field can either be length or type but not both)
<b>Data</b>	46-1500	The Network layer data.
<b>FCS</b>	4	Stores the CRC value which is used to check for errors in transmission.

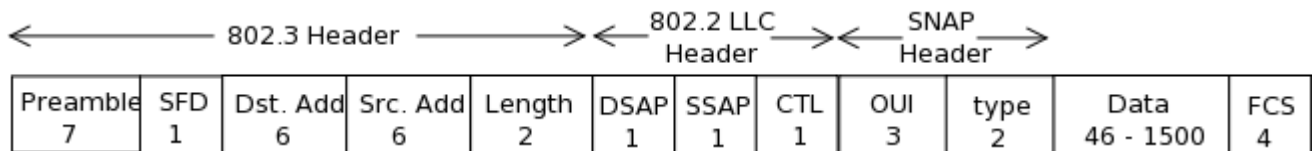
**Table 1-2 Frame Fields**

## TUN MIN OO {BE-IT} Routing & Switching 200-120

The Length/Type field is something you need to understand a little more about. The type field is very important because it tells the receiving end about the protocol whose data is contained in the frame. If the value of the field is less than a hex value of 0600 (decimal value 1536), it signifies that the field is used as a length field in that frame. For cases where this field is used as a length field, either one or two additional headers are added after the Ethernet 802.3 header, but before the layer 3 header. When IP packets are being carried, the Ethernet frame has the following two additional headers:

- An IEEE 802.2 Logical Link Control (LLC) header.
- An IEEE Subnetwork Access Protocol (SNAP) header.

Figure 1-20 shows an Ethernet frame with these two additional headers.



**Figure 1-20** 802.3 Frame with LLC and SNAP header



**Exam Alert:** It is not necessary to remember the fields of the frame. Just remember why LLC and SNAP headers are used for your CCNA exam.

## Ethernet at the Physical Layer

Ethernet was originally implemented by a group comprised of Digital, Xerox and Intel (DIX). IEEE then took over and created the 802.3 standard. This was a 10Mbps Ethernet that used co-axial cables.



**Exam Alert:** Ethernet is used to describe the family of standard that includes FastEthernet, Gigabit Ethernet etc. It is also used to describe the 10Mbps variant also which is simply noted as Ethernet.

IEEE then extended the 802.3 committee to two new committees known as the 802.3u (FastEthernet) and 802.3ab (Gigabit Ethernet on category 5 cable). Then it created another committee known as the 802.3ae (10Gbps over fiber and co-axial).

On the other hand the *Electronics Industries Association and the newer Telecommunication Industries Alliance (EIA/TIA)* is the standards body that creates the physical layer specifications for Ethernet. It specifies that a **registered jack (RJ) connector** with a 4 5 wiring sequence on an **unshielded twisted-pair (UTP)** cabling should be used with Ethernet. This cable comes in categories where higher category has less of the following two problems associated with them:

- **Attenuation** – This is the loss of signal strength as it travels the length of the cable. It is measured in decibels.
- **Crosstalk** – This is the unwanted signal interference from adjacent pairs in the cable.

What this means is that category 5 cable has lesser attenuation and crosstalk than category 3 cables.

Now that you know about the standards bodies involved and what they have done, it is time to look at the various Ethernet standards. Table 1-3 lists the original 3 standards. Remember that each standard is different in terms of Speed, Cable and the Maximum Length of cables.

Name	Speed	Cable Type	Max Cable length	Connector	Description
<b>10Base2</b>	10Mbps	Coaxial	185 meters	AUI	Known as <i>thinnet</i> , it can support up to 30 hosts in a single segment. A single collision domain across the network.
<b>10Base5</b>	10Mbps	Coaxial	500 meters	AUI	Known as <i>thicknet</i> , it can support up to 100 users in a single segment. A single collision domain across the network.
<b>10BaseT</b>	10Mbps	UTP	100 meters	RJ45	The first standard to use UTP cable with RJ45. A single host can be connected to a segment or wire. It required use of hubs to connect multiple hosts.

**Table 1-3 Original Ethernet Standards**



Table 1-4 shows the extended Ethernet Standards.

Name	Speed	Cable Type	Maximum Cable Length	Connector
<b>100BaseTX (IEEE 802.3u)</b>	100 Mbps	UTP cat. 5, 6 or 7 two-pair wiring	100 meters	RJ45
<b>100BaseFX (IEEE 802.3u)</b>	100Mbps	Multimode Fiber	412 meters	ST or SC connector
<b>1000BaseCX (IEEE 802.3z)</b>	1000Mbps	Copper twisted pair called twinax	25 meters	DE-9 or 8P8C
<b>1000BaseSX(IEEE 802.3z)</b>	1000Mbps	Multimode Fiber	220 meters	ST or SC connector
<b>1000BaseLX(IEEE 802.3z)</b>	1000Mbps	Single mode Fiber	5km	ST or SC connector
<b>1000BaseT(IEEE 802.3ab)</b>	1000Mbps	Cat 5 UTP	100 meters	RJ45

**Table 1-4 Extended Ethernet Standards**

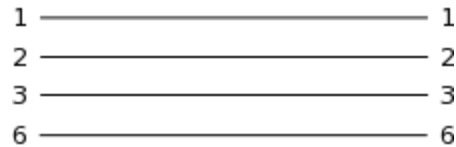
## Ethernet Cabling

When connecting different kinds of devices to each other, different kinds of cabling is used. The following three types of Ethernet cablings exist:

- Straight-through cable (a normal patch cable)
- Crossover cable
- Rolled cable

The three cabling types are discussed below:

**Straight-Through** – A UTP cable has 8 wires. A straight-through uses 4 out of these 8 wires. Figure 1-21 shows the configuration of the wire on both ends in a straight-through cable. Notice that only wires 1, 2, 3 and 6 are used and they connect straight to corresponding number on the other end.

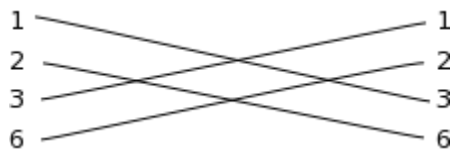


**Figure 1-21** *Wire configuration in Straight-Through cable*



**Note:** If you are wondering why the wire configuration is important remember that the transmitter on one end needs to connect to the receiver on the other end. If wiring configuration is incorrect, bits sent from one end will not be received at the other end.

**Crossover** – Crossover cable also uses the same four wires that are used in straight-through cable but different pins are connected here. Figure 1-22 shows the configuration of the wires in a crossover cable.



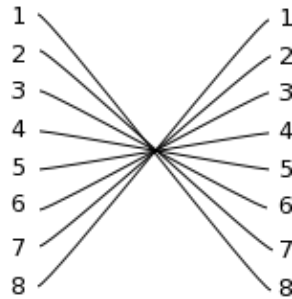
**Figure 1-22** *Wire configuration in Crossover cable*

Crossover cable is used to connect:

- Host to Host
- Switch to Switch
- Hub to Hub
- Switch to Hub
- Router to a host

Any easy way to remember this is that similar devices are connected to each other using crossover cables.

**Rolled Cable** – A rolled cable cannot be used for any Ethernet connection. It is used for connecting to a router's or a switch's console port from your host's serial communication (com) port. Every Cisco router and switch has a console port that is used for initial configuration. All 8 wires are used in this cable and each wire connects to the opposite number on the end (1 to 8, 2 to 7, 3 to 6 etc). Figure 1-23 shows the wire configuration.



**Figure 1-23** Wire configuration in Crossover cable



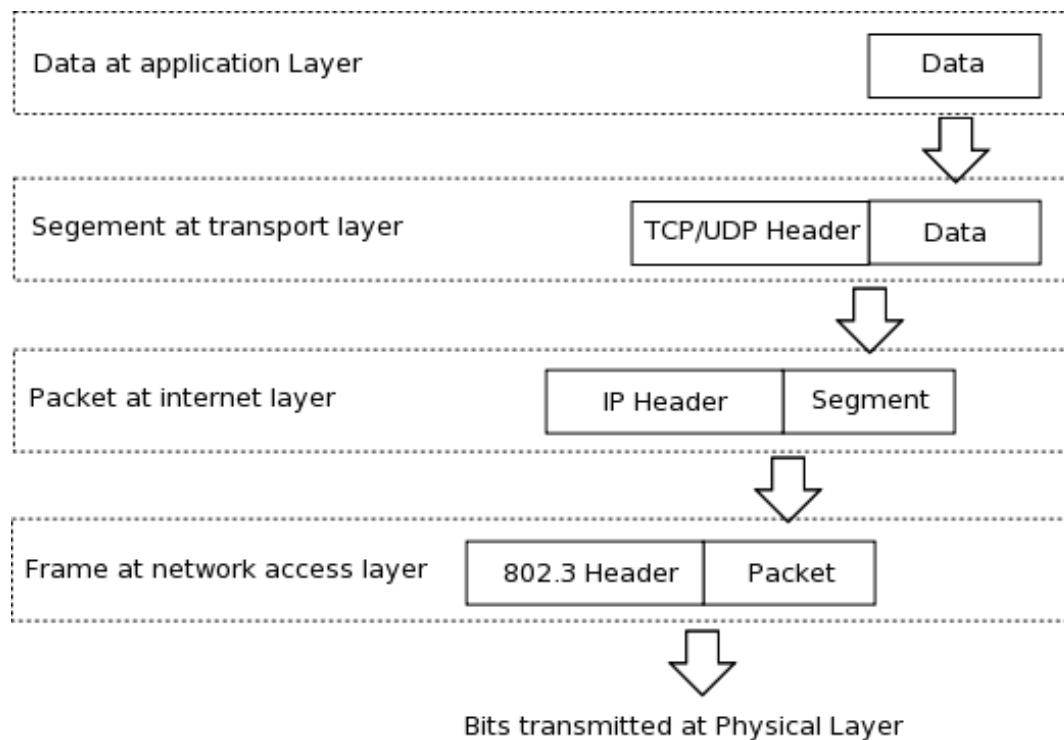
**Exam Alert:** Cable types and where they are used is a very important topic not only for the CCNA Exam as you will see questions on it, but also for your networking career as well.

### Data Encapsulation in TCP/IP Model

The last thing you need to know about TCP/IP model is the Data encapsulation process and PDUs. As in case of the OSI reference model, the data is encapsulated in a header (and trailer in case of Network layer) to create a Protocol Data Unit (PDU) and is passed down to the next layer. Though you are aware of the process, you must know the names of each layer's PDU. The PDU in TCP/IP model are:

- Transport Layer -> Segment
- Internet Layer -> Packet
- Network Access Layer -> Frame
- 

Figure 1-24 shows the encapsulation process in TCP/IP model.



**Figure 1-24** *Data encapsulation in TCP/IP Model*

♦♦♦♦♦ GO TO 1-6 ♦♦♦♦♦

### **1-6 Cisco 3 Layer Model**

In a large organization it is common to see large and complicated networks consisting of many locations, devices, services, and protocols. It can be cumbersome to manage and troubleshoot such networks. In addition to that as technologies evolve, the network has to evolve also. Making changes to a complex network is often difficult. Cisco with its years of experience in network equipment as well as managing its own network has defined a Three-layer hierarchical model. This model provides a hierarchical and modular method of building networks that makes it easy to implement, manage, scale and troubleshoot networks.

The model breaks an internetwork down to the following three layers:

- The Core layer
- The Distribution layer
- The Access layer

These layers are logical and not physical. They have specific functions in an internetwork which are discussed below:

**The Core Layer** – This layer is the backbone of an internetwork. It is the simplest yet the most critical layer whose sole function is to transport large amount of data fast. It gets data from the distribution layer and sends it back to the distribution layer after transportation. Speed and fault tolerance are the two major requirements of this layer because it has to transport large amount of data and any fault at this layer will impact every user. Considering the functions of this layer, the following should be avoided at this layer:

- Any thing that can slow down the traffic. For example, packet filtering, inter-VLAN routing etc.
- Direct user connections
- Direct server connections
- Complex service policies

While designing the core, the following should be kept in mind:

- Routing protocol should have low convergence time.
- Network Access layer technologies should be fast with low latency
- Redundancy should be built into this layer.

**The Distribution Layer** – This layer acts as an interface between the Core and the Access layers. The primary function of the distribution layer is to provide routing, filtering, and WAN access and to determine how packets can access the core, if needed. Path determination is the most important function at the layer. It has to select the fastest way an access request can be completed. This layer also acts as the convergence point for all access layer switches. Hence it is generally the best place to apply most of the policies. The following are generally done at this layer:

- Routing between subnets and VLANs and route distribution between routing protocols
- Implementation of security policies, including firewalls, address translations, packet filtering, etc.
- Breaking broadcast domains

**The Access Layer** – This layer is the edge of the network where wide variety of devices such as PCs, printers, iPads etc. connects to the network. Common resources needed by users are available at this layer while access request to remote resources are sent to the distribution layer. This layer is also known as the *desktop layer*. The following are generally done at this layer:

- Access control and policies in addition to what exists in the distribution layer.
- Dynamic configuration mechanisms
- Breaking collision domains
- Ethernet switching and static routing

-----\*\*\*\*\*-----

◆◆◆◆◆ GO TO 1-7 ◆◆◆◆◆

### **1-7Summary**

Though this chapter was long, it helped lay the foundation of your CCNA networking knowledge. The importance of understanding every topic in this chapter cannot be stressed enough. I would strongly suggest going through the chapter again to reinforce the basics.

The chapter started off with the importance of networks, basic network devices and network types and collision and broadcast domains.

Then the seven-layered OSI model was discussed. It is important to remember the functions of all the layers and how they map to the TCP/IP model. Remember that hubs work at Physical Layer, switches at Data-Link Layer and routers at the Network Layer of the OSI model.

The chapter then covered a long discussion on the TCP/IP model and its many protocols. Remember that TCP/IP and Ethernet form a major part of the CCNA exam and have a few chapters dedicated to them.

Lastly, the chapter covered the Cisco three-layer hierarchical model and how it is designed to help implement and manage a complex network.

The next chapter looks at IP addressing. Before heading to it, we suggest you review the CCNA **Exam Alerts** scattered through this chapter to recap the various important concepts.

**Ending Chapter 1. &Go to Chapter-2**

## **Chapter 2 – IP Addressing and Subnets**

Chapter 1 introduced you to the various layers of the TCP/IP model. The CCNA exam is almost entirely about the Internet and the Network Access layer. So this chapter will cover one of the most important subjects of networking – IP Addresses. As you already know, each host in the network has a logical address called the IP address. This address helps in routing packets from source to destination across internetworks. This chapter delves deep into IP addresses, subnet mask, subnetting and Variable Length Subnet Mask (VLSM). Finally this chapter looks at some troubleshooting techniques that are used to solve IP address related problems. The two current versions of IP addresses in use today are IPv4 and IPv6. This chapter focuses on IPv4. IPv6 is discussed in Chapter 12.

### **2-1 IP Addresses – Composition, Types and Classes**

Before heading deeper into IP addresses, you should be aware of the following terms

- **Bit** – A bit is a single digit with a value of 0 or 1.
- **Byte** – A byte is composed of 8 bits.
- **Octet** – An octet is also made up of 8 bits. Throughout this chapter the terms byte and octet are interchangeable.
- **Network Address** – This refers to a remote network in terms of routing. All hosts in the remote network fall within this address. For example, 10.0.0.0, 172.16.0.0 and 192.168.1.0
- **Broadcast Address** – This is the address used to send data to all hosts in a network. The broadcast address 255.255.255.255 refers to all hosts in all networks while an address such as 192.168.1.255 refers to all hosts in a particular network.

An IP address is 32 bits in length. To make the address easier to read, it is divided into four sections of 8 bits each divided by a period. Each section is therefore, 1 byte (also called octet) long. To further make it easier to read and remember, the binary numbers are converted to decimal. For example, an IP address such as 11000000100000000000110000000001 is divided to make it 11000000.10000000.00001100.00000001. When this address is converted to decimal, it will become 192.128.12.1. This format of IP address is called the **dotted decimal** format. Some applications also convert the address to hexadecimal format instead of decimal format. However this is not commonly seen and as far as the CCNA exam is concerned, you need to only work with the dotted decimal format.

## TUN MIN OO {BE-IT} Routing & Switching 200-120

Topics in this chapter require binary to decimal conversions. Table 2-1 shows the decimal value of each bit location in a byte. To easily convert from binary to decimal, add up the decimal value corresponding to the bit place that is “on” (1). For example, a binary value of 10110000 can be easily converted to decimal by adding the decimal value of each bit that is 1. That gives us  $128+32+16 = 176$ .

Table 2-2 shows the decimal value for the most common binary numbers you will encounter in this chapter.

128	64	32	16	8	4	2	1
-----	----	----	----	---	---	---	---

**Table 2-1** *Decimal Value for each bit place in a byte*

<i>Binary Value</i>	<i>Decimal Value</i>
10000000	128
11000000	192
11100000	224
11110000	240
11111000	248
11111100	252
11111110	254
11111111	255

**Table 2-2** *Decimal Values for common binary numbers*



An IP address does not only represent the host address. In fact it represents the network where the host resides and the host it self. In effect, the IP address consists of two parts:

- i. **The Network component** – Defines network (or **subnet**), in an internetwork, the host resides in.
- ii. **The Host component** – Defines the host itself in the network.

Each combination of the network component and the host component should be unique in the entire Internetwork. To make it easy to identify which portion of the address is network component and which one is the host component, addresses are broken down into 5 classes discussed below:

- **Class A** – The first byte (8 bits) is the network component and the remaining three bytes (24 bits) are host component (network.host.host.host). This class is for an internetwork with small number of networks and large number of hosts per network.
- **Class B** – The first two bytes (16 bits) are the network component and the remaining three bytes are host components (network.network.host.host). This class bridges the gap between Class A and Class C by providing for medium number of networks with medium number of hosts.
- **Class C** – The first three bytes (24 bits) are the network component and the last byte (8 bits) is the host components (network.network.network.host). This class provides for large number of networks with fewer hosts per network.
- **Class D** – Used for multicasting.
- **Class E** – Reserved addresses

In a binary address the first 5 bits of the address and the first octet in a dotted decimal address shows the class of address. Table 2-3 shows the first 5 bits and the first octet range of each class of address.

Class	First 5 bits in binary	First Octet range
A	0xxxx	0-127 (actually 1-126 because 0 and 127 are reserved)
B	10xxx	128-191
C	110xx	192-223
D	1110x	224-239
E	1111x	240-254

**Table 2-3 Address range for different classes of address**

Notice that first few bits in each class have a fixed value. For example a class A address should have the first bit set to 0. Similarly class C should have first 2 bits set to 1 and the third bit set to 0. Another point to note is that though the class A range is from 0 to 127, the address 0.0.0.0 is reserved to mean “any network” and 127.0.0.1 is reserved as a loopback address which refers to the host itself. So the class A network is restricted to the 1-126 range.



**Exam Alert:** Class of addresses and their address range is a very important topic. You will have to remember the range associated with each class.

Before moving ahead, spend some time to figure out the class of some addresses given below. Also try to figure out which portion is the network and which portion is the host part:

- a) **9.140.2.87** – This is a Class A address because the first octet lies in 1-126 range. 9 is the network part while 140.2.87 is the host part because class A addresses have a network.host.host.host format.
- b) **172.30.4.190** – This is a Class B address because the first octet lies in 128-191 range. 172.30 is the network part while 4.190 is the host part because class B addresses have a network.network.host.host format.
- c) **194.144.5.10** – This is a Class C address because the first octet lies in the 192-223 range. 194.144.5 is the network part while 10 is the host part because class C addresses have a network.network.network.host format.
- d) **45.22.187.1** – This is again a class A address with 45 being the network part and 22.187.1 being the host part.

*Some IP address such as 127.0.0.1 have a special meaning. Table 2-4 lists such addresses and what they represent.*

Address	What it represents	Where can it be used
<b>Network address of all 0s</b>	Represents “this network”. For example 0.0.0.120	For sending broadcast messages to the network.
<b>Network address of all 1s</b>	Represents “all networks”.	For sending broadcast messages to all networks.
<b>Node address of all 0s</b>	Represents a network address or all hosts in the network. Example 10.0.0.0 or 172.16.0.0	Routers route traffic based on network address.
<b>Node address of all 1s</b>	Represents all hosts in a network. Also called the broadcast address. Example 172.16.255.255 or 192.168.10.255	Used to send broadcasts to all hosts in a network.
<b>Entire address of 0s</b>	Represents “any network”.	Used by routers to designate the default route.
<b>Entire IP set to all 1s.</b>	Represents all hosts in network.	Used to send broadcast messages
<b>127.0.0.1</b>	Represents the loopback address which is essentially the host itself	To send traffic from the host to itself. If you want to connect to a webserver running on the host itself, you will use this address in the browser.

**Table 2-4 Reserved IP addresses**

**Exam Alert:** It is important to remember that if all host bits in an address are set to 0 then it is a network address. On the other hand if all host bits are set to 1 then it is a broadcast address. These addresses cannot be assigned to a host.

◆◆◆◆◆ GO TO 2-2◆◆◆◆◆

## **2-2 Private and Public IP addresses**

As you know already, every host on a network requires a unique IP address. This is easily manageable in a small network but not a network as large as the Internet. The Internet Assigned Numbers Authority (IANA) is responsible for managing and distributing IP addresses. The IANA has created 5 address registrars in five locations of the world. ISPs and large organizations purchase the addresses from these registrars. The end user in turn gets the IP address from the ISP. These purchasable IP addresses are called **public addresses** and are routable on the Internet. Every host on the Internet has one of these addresses, in theory.

The IANA also designated a range of addresses in class A, B and C for use in private networks. These addresses can be used by anyone within their network without any required permission but these addresses are not routable on the Internet. Your ISP or your organization usually assigns you one of these addresses and later **translates** it to a public address when you want to get out to the Internet. The designated ranges for private IP addresses are:

- **Class A – 10.0.0.0 to 10.255.255.255 (1 network)**
- **Class B – 172.16.0.0 to 172.31.255.255 (16 networks)**
- **Class C – 192.168.0.0 to 192.168.255.255 (256 networks)**

*Address translation and private IP addresses are discussed in detail in Chapter 9.*



**Exam Alert:** It is very important to remember the range of private IP addresses as you will more than likely see a question about them on your CCNA exam.

◆◆◆◆◆ GO TO 2-3◆◆◆◆◆

## **2-3 Subnetting**

In case of class A and B IP addresses, each of them provides for a large number of hosts. For class A, the total numbers of hosts available are  $2^{24}-2$  or 16,777,216 hosts (class A has 24 bits available for host component and each bit can have two values – 0 and 1. Out of the total value one address is for network address and the other for broadcast. So two addresses are deducted). Similarly a Class B addresses provides for  $2^{16}-2$  or 65,534 hosts. In the first chapter you learned about disadvantages of large networks and why it becomes necessary to divide them into smaller networks joined by routers. So creating a network with total number of hosts allowed for class A or B addresses will cause a lot of problems. Meanwhile creating small networks with class A or B addresses will waste a lot of addresses.

To overcome this problem with class based addressing, subnetting was introduced. Subnetting allows you to borrow some host bits and use them to create more networks. These networks are commonly called **subnets** and are smaller in size. But since each network has a network address and a broadcast address, some addresses get wasted.

To further understand how subnetting is useful consider a Class C address. Each class C address has  $2^8-2$  or 254 host addresses available. If you wanted 2 networks with 100 addresses and used 2 class C networks, you would waste 308 addresses. Instead of using two class C networks, you can subnet one to provide you two networks of 126 addresses each. This way lesser number of addresses would be wasted.

While some of the benefits of subnetting are discussed above, the following list discusses all the benefits associated with it:

- **Reduced broadcasts** – While broadcasts are necessary, too many of them can bring down a network and the number of broadcasts is proportionate to the size of the network. So subnetting a network to smaller subnetworks, helps reduce broadcasts since routers do not forward broadcasts.
- **Increased Network Performance** – The direct result of reduced broadcasts is a network that has more bandwidth available to the hosts. More bandwidth and lesser hosts result in a better performance of the network.
- **Easier Management** – Managing and troubleshooting a large network is cumbersome and difficult. Subnetting breaks a network into smaller subnetworks, making it easier to manage each of them.
- **Scalability** – A single large network spanning a large geographical location will be more difficult and costlier to manage. WAN links connecting different locations are costly and having

broadcasts choking the network can result is wasted money. Hence breaking down a large network makes is easier to scale a network across geographical locations.

Now that you understand the concept and benefit of subnetting, consider the problem that arises with it. In case of class based subnetting, the first octet of the dotted decimal address tells which part of the address is the network component and which one is the host component. But when host bits are borrowed for subnetting, the class based boundaries do not apply and it is not possible to say which bits are network bits. To overcome this, a third component of IP addresses were added. These are called the **subnet masks**.

Subnets masks, like IP addresses, are 32 bit long. The value of subnet mask represents which bits of the IP address are network components and which are host component. A value of 1 in a subnet mask shows that the corresponding bit in the IP address is a network component while a value of 0 shows that the corresponding bit is a host component. The following examples will help clarify this further:

- i. An IP address of 192.168.10.1 with a subnet mask of 255.255.255.0 (11111111.11111111.11111111.00000000) shows that the first three octets of the IP address are the network component while the last octet is the host component.
- ii. An IP address of 172.16.100.1 with a subnet mask of 255.255.128.0 (11111111.11111111.10000000.00000000) shows that one bit from the third octet has been borrowed from the host component. Hence the network component is now 17 bits long instead of the default 16 bit in a class B address.
- iii. An IP address of 10.1.1.1.1 with a subnet mask of 255.255.0.0 (11111111.11111111.00000000.00000000) shows that the entire second octet has been borrowed from the host component and now the network component is 16 bits long instead of the default 8 bit of a class A address.

One restriction that applies to subnet masks is that all network bits (1) and all host bits (0) should be contiguous. So a subnet mask of 11001100.11110000.11110000.00001111 is not valid because the network and host bits are not contiguous. Table 2-5 shows the valid subnet mask values is an octet.

Binary Value	Decimal Value
00000000	0
10000000	128
11000000	192
11100000	224
11110000	240
11111000	248
11111100	252
11111110	254
11111111	255

**Table 2-5** *Valid subnet mask values in an octet*

Subnets masks are commonly represented in two ways:

- Dotted Decimal – 10.1.1.1 255.255.0.0
- Classless Inter-Domain Routing (CIDR) notation – 10.1.1.1/16



**Exam Alert:** It is very important to be able to understand subnet masks with both the dotted decimal as well as the CIDR format. Also remember that any mask not given in Table 2-5 is not valid for an octet.

By now you may have figured out that the default subnet mask of class A is 255.0.0.0 or /8, the default mask of class B is 255.255.0.0 or /16 and the default mask of class C is 255.255.255.0 or /24. Table 2-6 shows the default masks of each class. These default masks cannot be changed. For example, you cannot use a mask of 255.255.0.0 for a class C address. If you try to use an invalid mask such as this, every device will produce an error. For each class, the minimum mask is the default mask and it cannot

be reduced. Class A has to have a minimum mask of 255.0.0.0, class B has to have a minimum mask of 255.255.0.0 and class C has to have a minimum mask of 255.255.255.0.

<b>Class</b>	<b>Format</b>	<b>Default Subnet Mask</b>
<b>A</b>	<b>network.host.host.host</b>	<b>255.0.0.0</b>
<b>B</b>	<b>network.network.host.host</b>	<b>255.255.0.0</b>
<b>C</b>	<b>network.network.network.host</b>	<b>255.255.255.0</b>

**Table 2-6** *Default Subnet masks*

Remember that an IP address without mask or a mask without IP address does not mean anything. A mask of /24 does not mean that the address is a class C address. Even a class A or class B address can have a mask of /24 after subnetting. Table 2-7 provides a list of dotted decimal subnet mask value and the corresponding CIDR value.

<b>Dotted Decimal Value</b>	<b>CIDR notation</b>
<b>255.0.0.0</b>	<b>/8</b>
<b>255.128.0.0</b>	<b>/9</b>
<b>255.192.0.0</b>	<b>/10</b>
<b>255.224.0.0</b>	<b>/11</b>
<b>255.240.0.0</b>	<b>/12</b>
<b>255.248.0.0</b>	<b>/13</b>
<b>255.252.0.0</b>	<b>/14</b>
<b>255.254.0.0</b>	<b>/15</b>
<b>255.255.0.0</b>	<b>/16</b>
<b>255.255.128.0</b>	<b>/17</b>
<b>255.255.192.0</b>	<b>/18</b>
<b>255.255.224.0</b>	<b>/19</b>
<b>255.255.240.0</b>	<b>/20</b>
<b>255.255.248.0</b>	<b>/21</b>
<b>255.255.252.0</b>	<b>/22</b>



<b>255.255.254.0</b>	/23
<b>255.255.255.0</b>	/24
<b>255.255.255.128</b>	/25
<b>255.255.255.192</b>	/26
<b>255.255.255.224</b>	/27
<b>255.255.255.240</b>	/28
<b>255.255.255.248</b>	/29
<b>255.255.255.252</b>	/30

**Table 2-7** *Subnet Mask values*

Before moving to actual subnetting, you need to remember the powers of 2 i.e. the value of 2 multiplied as many times as the given exponent. For example  $2^4 = 2 \times 2 \times 2 \times 2 = 16$ . Table 2-8 lists the first 14 values. It is not necessary to remember them all, but do remember that each value is twice the previous one. The more you remember these values, the easier it will be to subnet on your CCNA exam

<b>Exponent</b>	<b>Value</b>
<b><math>2^1</math></b>	2
<b><math>2^2</math></b>	4
<b><math>2^3</math></b>	8
<b><math>2^4</math></b>	16
<b><math>2^5</math></b>	32
<b><math>2^6</math></b>	64
<b><math>2^7</math></b>	128
<b><math>2^8</math></b>	256
<b><math>2^9</math></b>	512
<b><math>2^{10}</math></b>	1024
<b><math>2^{11}</math></b>	2048
<b><math>2^{12}</math></b>	4096
<b><math>2^{13}</math></b>	8192
<b><math>2^{14}</math></b>	16384

**Table 2-8 Powers of Two**

Now that you know what subnetting is and how subnet masks are used, it is time to create subnets. When planning to subnet, you need to know three things:

1. Total number of subnets that you need
2. Total number of hosts per subnet that you need
3. Available network and subnet mask (which will be subnetted)

Armed with answers to this, you need to find the following:

- ✓ Subnet Mask to be used across the network
- ✓ Valid subnets
- ✓ Network address for each subnet
- ✓ Broadcast address for each subnet
- ✓ Valid host addresses in each subnet.

For this section I will take a sample requirement of 8 networks with 30 hosts each with one class C network of 192.168.10.0 255.255.255.0 available. Now that you have the requirement, first thing you need to find is the new subnet mask that can satisfy the requirement. To find the subnet mask, follow the steps given below:

- I. Find the exponent of 2 whose value is more than or equal to the number of subnets required. Lets call this  $2^{sn}$ . For our example, we need 8 subnets and  $2^3$  equals to 8. So our  $2^{sn}$  is  $2^3$ .
- II. Find the exponent of 2 whose value minus 2 is more than or equal to the maximum number of hosts required in a subnet. Lets call this  $(2^h - 2)$  For our example, we need a maximum of 30 hosts in a subnet and  $2^5 - 2$  gives us 30 hosts per subnet.
- III. Make sure  $sn + h$  from the above two steps does not exceed the number of host bits available in the network available. If the sum of  $sn$  and  $h$  exceed the available host bits then you will require another network of the same class or a network of a higher class. In our example we have 8 bits of host addresses available in 192.168.10.0 255.255.255.0 network. Our  $sn+h$  is  $3+5$  that gives us 8.
- IV. Convert the available mask to the CIDR notation and add  $sn$  to it to get the new subnet mask. For our example the mask 255.255.255.0 can be converted to /24. On adding 3 we get a mask of /27. Converting from /27 to the dotted decimal format is easy. /24 is 255.255.255.0 or 11111111.11111111.11111111.00000000. /27 will be 11111111.11111111.11111111.11100000. You need not worry about the first 3 octets since they are already known to be 255.255.255. For the last octet add the decimal value for each network bit. In our case it will be  $128+64+32$

= 224. So the new subnet mask is 255.255.255.224. Table 2-7 also provides a list of dotted decimal and networking bits value.

The most difficult part is now over. To find the rest of the 4 answers, follow the steps given below:

- I. Valid subnets – To find the valid subnets deduct the interesting octet value from 256. Interesting octets are those octets that have host bits. Available subnets will be in multiples of the resultant value up to 256. In our case the fourth is the interesting octet. Deducting 224 from 256 gives us 32. So the available subnets are 0, 32, 64, 96, 128, 160, 192, 224.
- II. Network Address of each subnet – The network address is the very first address of each subnet. So for our valid subnets, the network address would be 192.168.10.0, 192.168.10.32, 192.168.10.64, 192.168.10.96, 192.168.10.128, 192.168.10.160, 192.168.10.192 and 192.168.10.224



**Exam Alert:** Sometime back Cisco used to discard the first and the last subnet, also called **subnet zero**. So the number of subnets used to be  $2^n - 2$ . Starting IOS version 12.0 the **ip subnet-zero** command is enabled by default and in Cisco exams the first and last subnets are considered unless specified otherwise. Be on the lookout for questions on your CCNA exam that ask you not to consider subnet zero. In such cases, leave out the first and the last subnet. To fully understand how the command affects the calculation, consider a Class C network with a mask of /26. It will give you subnets 0, 64, 128 and 192 if subnet-zero is allowed, else it will only give you subnets 64 and 128.

- III. Broadcast Address of each subnet – The last address of a subnet is the broadcast address. Simply deduct 1 from the next network address to find the broadcast address of a subnet. For our example subnets the valid broadcast addresses are:

Network Address	Broadcast Address
192.168.10.0	192.168.10.31
192.168.10.32	192.168.10.63
192.168.10.64	192.168.10.95
192.168.10.96	192.168.10.127
192.168.10.128	192.168.10.159
192.168.10.160	192.168.10.191

192.168.10.192	192.168.10.223
192.168.10.224	192.168.10.255

- IV. Valid hosts addresses in each subnet – For every subnet, the valid host addresses lie between the network address and the broadcast address. For our example, the valid host addresses for each subnet are:

Network Address	Valid Host addresses	Broadcast Address
192.168.10.0	192.168.10.1 – 30	192.168.10.31
192.168.10.32	192.168.10.33 – 62	192.168.10.63
192.168.10.64	192.168.10.65 – 94	192.168.10.95
192.168.10.96	192.168.10.97 – 126	192.168.10.127
192.168.10.128	192.168.10.129 – 158	192.168.10.159
192.168.10.160	192.168.10.161 – 190	192.168.10.191
192.168.10.192	192.168.10.193 – 222	192.168.10.223
192.168.10.224	192.168.10.225 – 254	192.168.10.255



**Exam Alert:** Subnetting is one of the most important topics in the CCNA exam. Subnetting related questions will not be straight forward like what you learned just now. Mostly you would be given an IP address with a subnet mask and you will need to find out if it is a host, subnet or broadcast address. In following examples review how to approach such questions.

In the following sections, you will encounter variations of subnetting questions. For all of them the process is similar to what you just learned. The steps you need to follow are summarized below:

- i. Find the interesting octet in the given subnet mask. Remember that the octet with a value of less than 255 will be the interesting octet.
- ii. Deduct the value of interesting octet from 256 to find the increment by which the network numbers are increasing. These are also your subnet addresses.
- iii. Write down the subnet address and broadcast address for each subnet
- iv. Write down the host addresses of each subnet
- v. Once you have all the above information, you will find the answer to the given question.

### Subnetting Class C Addresses

Subnetting technique remains the same irrespective of the class of address. The difference that the class makes is the number of bits available for subnetting. Class C starts with a mask of /24 and can have a maximum mask of /30. We cannot use /31 or /32 because atleast 2 hosts bits are required for the network and broadcast addresses and /31 and /32 give us 1 and zero host bits respectively. In the examples below, you get to practice subnetting class C addresses.

#### Subnetting Class C Address – Example #1

Problem: Is 192.168.1.193/26 a host address?

Solution:

- i. Converting /26 to dotted decimal format gives 255.255.255.192. The fourth octet is the interesting octet.
- ii. Deducting 192 from 256 gives us 64. So the subnet addresses are 0,64,128 and 192
- iii. The network address and broadcast address are:

Network Address	Broadcast Address
192.168.1.0	192.168.1.63

## TUN MIN OO {BE-IT} Routing & Switching 200-120

<b>192.168.1.64</b>	192.168.1.127
<b>192.168.1.128</b>	192.168.1.191
<b>192.168.1.192</b>	192.168.1.255

iv. The host addresses for each of the subnets are:

Network Address	Host Addresses	Broadcast Address
<b>192.168.1.0</b>	192.168.1.1-62	192.168.1.63
<b>192.168.1.64</b>	192.168.1.65-126	192.168.1.127
<b>192.168.1.128</b>	192.168.1.129-190	192.168.1.191
<b>192.168.1.192</b>	192.168.1.193-254	192.168.1.255

v. The given address, 192.168.1.193 is a host address in the last subnet.

### Subnetting Class C Address – Example #2

Problem: What is the network and broadcast address for the subnet to which the address 192.168.1.228/28 belongs?

Solution:

- I. Converting /28 to dotted decimal format gives 255.255.255.240. This shows that the fourth octet is the interesting octet.
- II. Deduction 240 from 256 gives us 16. So the subnet addresses are 0, 16, 32, 48, 64 ... 208, 224, 240.
- III. The network and broadcast address for the subnets are:

Network Address	Broadcast Address
-----------------	-------------------

## TUN MIN OO {BE-IT} Routing & Switching 200-120

<b>192.168.1.0</b>	192.168.1.15
<b>192.168.1.16</b>	192.168.1.31
<b>192.168.1.32</b>	192.168.1.47
<b>192.168.1.48</b>	192.168.1.63
<b>192.168.1.64</b>	192.168.1.79
<b>192.168.1.208</b>	192.168.1.223
<b>192.168.1.224</b>	192.168.1.239
<b>192.168.1.240</b>	192.168.1.255

IV. The host addresses of each subnet are:

Network Address	Host Addresses	Broadcast Address
<b>192.168.1.0</b>	192.168.1.1- 192.168.1.14	192.168.1.15
<b>192.168.1.16</b>	192.168.1.17- 192.168.1.30	192.168.1.31
<b>192.168.1.32</b>	192.168.1.33- 192.168.1.46	192.168.1.47

## TUN MIN OO {BE-IT} Routing & Switching 200-120

<b>192.168.1.48</b>	192.168.1.49- 192.168.1.62	192.168.1.63
<b>192.168.1.64</b>	192.168.1.65- 192.168.1.78	192.168.1.79
<b>192.168.1.208</b>	192.168.1.209- 192.168.1.222	192.168.1.223
<b>192.168.1.224</b>	192.168.1.225- 192.168.1.238	192.168.1.239
<b>192.168.1.240</b>	192.168.1.241- 192.168.1.254	192.168.1.255

- V. From the above table, you can see that the address 192.168.1.228 lies in the 192.168.1.224 subnet. The network address for this subnet is 192.168.1.224 and the broadcast address is 192.168.1.239.

### Subnetting Class C Address – Example #3



## TUN MIN OO {BE-IT} Routing & Switching 200-120

Problem: What type of address is 192.168.5.47/29? What is the network and broadcast address of the subnet that this address belongs to and how many host addresses are available in the subnet?

Solution:

- i. Converting /29 gives 255.255.255.248. This shows that the fourth octet is the interesting octet.
- ii. Deducting 248 from 256 gives us 8 so the subnets are 0, 8, 16, 24, 32, 40, 48...240,248
- iii. 192.168.5.47 lies in the 192.168.5.40 subnet and is the last address before the next subnet 192.168.5.48. This means that 192.168.5.47/29 is a broadcast address for the 192.168.5.40/29 subnet.
- iv. The network address for this subnet is 192.168.5.40 and the valid host address range is 192.168.5.41-192.168.5.46

### Subnetting Class B addresses

The process to subnet class B addresses is same as that used to subnet class C address. The difference is that you have more bits available for subnetting. Class B addresses start with a mask of /16 and can have a maximum mask of /30. One big difference when subnetting class B addresses is that you deal with large number of hosts per subnet and it becomes important to remember the Powers of Two table shown in Table 2-8. In the examples given below, you will practice subnetting class B addresses.

### Subnetting Class B address – Example #1

Problem: Is 172.16.98.45/19 a host address?

Solution:

1. Converting /19 to dotted decimal format gives us 255.255.224.0. The third octet is the interesting octet.
2. Deducting 224 from 256 gives 32. So the subnet addresses are 0, 32, 64, 96, 128, 160, 192, 224
3. The network address and broadcast address are:

Network Address	Broadcast Address
172.16.0.0	172.16.31.255
172.16.32.0	172.16.63.255

## TUN MIN OO {BE-IT} Routing & Switching 200-120

<b>172.16.64.0</b>	172.16.95.255
<b>172.16.96.0</b>	172.16.127.255
<b>172.16.128.0</b>	172.16.191.255
<b>172.16.192.0</b>	172.16.223.255
<b>172.16.224.0</b>	172.16.255.255

4. The host address range for each subnet is:

Network Address	Host Addresses	Broadcast Address
<b>172.16.0.0</b>	172.16.0.1-172.16.31.254	172.16.31.255
<b>172.16.32.0</b>	172.16.32.1-172.16.63.254	172.16.63.255
<b>172.16.64.0</b>	172.16.64.1-172.16.96.254	172.16.95.255
<b>172.16.96.0</b>	172.16.96.1-172.16.127.254	172.16.127.255
<b>172.16.128.0</b>	172.16.128.1-172.16.191.254	172.16.191.255
<b>172.16.192.0</b>	172.16.192.1-172.16.223.254	172.16.223.255
<b>172.16.224.0</b>	172.16.224.1-172.16.255.254	172.16.255.255

5. The address 172.16.98.45 is a host address in the 4<sup>th</sup> subnet.

### Subnetting Class B address – Example #2

Problem: What are the network and broadcast addresses for the subnet to which the address

172.19.251.100/23 belongs.

Solution:

1. Converting /23 to dotted decimal format gives us 255.255.254.0. This shows that the third octet is the interesting octet.
2. Deducting 254 from 256 gives us 2. So the subnet addresses are 0, 2, 4, 6, 8, 10...248, 250, 252, 254
3. The network, broadcast and valid host ranges for these subnets are:

Network	Host Addresses	Broadcast
---------	----------------	-----------

Address		Address
<b>172.19.0.0</b>	172.19.0.1- 172.19.1.254	172.19.1.255
<b>172.19.2.0</b>	172.19.2.1- 172.19.3.254	172.19.3.255
<b>172.19.4.0</b>	172.19.4.1- 172.19.5.254	172.19.4.255
<b>172.19.6.0</b>	172.19.6.1- 172.19.7.254	172.19.7.255
<b>172.19.8.0</b>	172.19.8.1- 172.19.9.254	172.19.9.255
<b>172.19.250.0</b>	172.19.250.1- 172.19.251.254	172.19.251.255
<b>172.19.252.0</b>	172.19.252.1- 172.19.253.254	172.19.253.255
<b>172.19.254.0</b>	172.19.254.1- 172.19.255.254	172.19.255.255

5. As you can see, the address 172.19.251.100/23 is a valid host address in the 172.19.250.0/23 subnet. The network address for this subnet is 172.19.250.0 and the broadcast address is 172.19.251.255.

### Subnetting Class B address – Example #3

Problem: You see that your PC has an IP address and subnet mask of 172.30.40.5/21. How many subnets can your network have? How many valid host addresses can each subnet have?

Solution:

1. Converting a /21 mask to dotted decimal format gives us 255.255.248.0.
2. Converting it to dotted binary format gives us 11111111.11111111.11111000.00000000. This shows that 5 bits have been borrowed for subnets and 11 bits are available for host addresses.
3. The borrowed 5 bits gives us  $2^5=32$  subnets.
4. The 11 host bits give us  $2^{11} = 2048$  addresses. Out of 2048, 2 addresses are reserved for host and broadcast addresses. So this leaves us with 2046 valid host addresses per subnet.

### Subnetting Class A addresses

The process to subnet class A addresses is the same as that you have used to subnet class C and B addresses. The big difference is the large numbers you can deal with while using masks such as /9. Class A addresses start with a mask of /8 and can have a maximum of /30 mask. In the examples below, you will practice subnetting class A addresses.

### Subnetting Class A address – Example #1

Problem: Is 10.127.255.254/9 a host address?

Solution:

## TUN MIN OO {BE-IT} Routing & Switching 200-120

1. Converting /9 to dotted decimal format gives 255.128.0.0. The second octet is the interesting octet.
2. Deducting 128 from 256 gives 128. So the subnet addresses are 0 and 128.
3. The network and broadcast address are:

Network Address	Broadcast Address
10.0.0.0	10.127.255.255
10.128.0.0	10.255.255.255

4. The host address range for the subnets are:

Network Address	Host Addresses	Broadcast Address
10.0.0.0	10.0.0.1-10.127.255.254	10.127.255.255
10.128.0.0	10.128.0.1-10.255.255.254	10.255.255.255

5. 10.127.255.254 is the last host address in the 1<sup>st</sup> subnet.



**Exam Alert:** A /30 or 255.255.255.252 is the highest mask which can be practically used in a network. It gives 2 host addresses and is ideal for point-to-point links in a network. Point-to-Point links are usually found in routers terminating WAN links.

### Subnetting Class A address – Example #2

**Problem:** This is a different kind of a problem. Your network number is 21.0.0.0. You need to have as many subnets as possible without exceeding 1000 subnets while at the same time having at least 500 hosts per subnet. What subnet mask would you use?

**Solution:**

Since 21.0.0.0 is a Class A network, the default mask is /8. So you have 24 bits of host addresses that can be borrowed for the subnetting. Looking back at Table 2-8, you will see that  $2^{10}$  gives us 1024 while  $2^9$  gives us 512. Since 1024 exceeds the given 1000 subnets, you will need to use  $2^9$ . This means 9 bits will be borrowed for the network part leaving the rest for the host part. The table below shows the default mask and the new mask after borrowing 9 bits:

Octets	1 <sup>st</sup> Octet	2 <sup>nd</sup> Octet	3 <sup>rd</sup> Octet	4 <sup>th</sup> Octet
--------	-----------------------	-----------------------	-----------------------	-----------------------

<b>Default mask</b>	11111111	00000000	00000000	00000000
<b>New mask</b>	11111111	<b>11111111</b>	10000000	00000000

The new mask of /17 will leave 15 bits for the host part which gives us much more than the required 500 hosts per subnet.

### Subnetting Class A address – Example #3

Problem: You have been given a network number of 10.0.0.0/8. You need to subnet it such that you have at least 8000 hosts per subnet and at least 2000 subnets. What subnet mask will you use?

Solution:

10.0.0.0/8 is a class A address with a default mask of /8. This leaves you with 24 bits for host addresses. So you need to find which multiples of 2 give us the required numbers. Looking back at Table 2-8, you will see that 2<sup>11</sup> gives us 2048 while 2<sup>13</sup> gives us 8192. This means you can borrow 11 bits for the network part, leaving 13 bits for the host part. The table below shows the default mask and the new mask in binary format:

Octets	1 <sup>st</sup> Octet	2 <sup>nd</sup> Octet	3 <sup>rd</sup> Octet	4 <sup>th</sup> Octet
<b>Default mask</b>	11111111	00000000	00000000	00000000
<b>New mask</b>	11111111	<b>11111111</b>	11100000	00000000

10.0.0.0/19 will give you 2048 subnets with 8192 host bits remaining. Each subnet will have a maximum of 8191 hosts, leaving 2 addresses for network and broadcast addresses.

### Subnetting Class A address – Example #4

Problem: What are the network and broadcast addresses for the subnet to which the address 10.212.10.50/12 belongs.

Solution:

1. Converting /12 to dotted decimal format gives us 255.240.0.0. This shows that the second octet is the interesting octet.
2. Deducting 240 from 256 gives us 16. This means that the valid subnets are 0, 16, 32, 48, 64...208, 224, 240
3. The network, valid host and broadcast addresses for these subnets are:

## TUN MIN OO {BE-IT} Routing & Switching 200-120

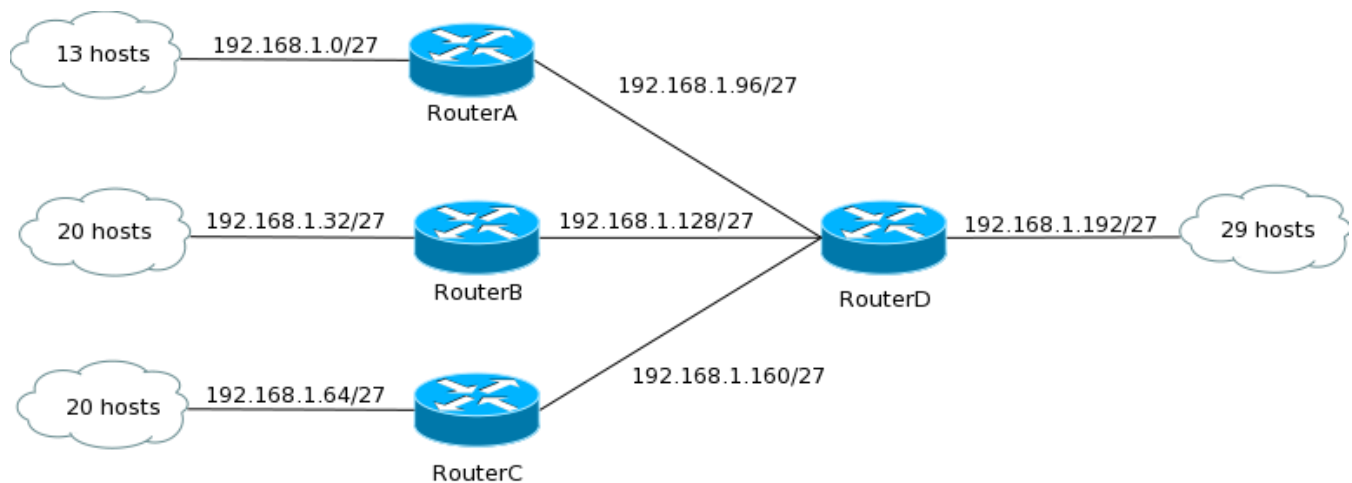
Network Address	Host Addresses	Broadcast Address
<b>10.0.0.0</b>	10.0.0.1-10.15.255.254	10.15.255.255
<b>10.16.0.0</b>	10.16.0.1- 10.31.255.254	10.31.255.255
<b>10.32.0.0</b>	10.32.0.1- 10.47.255.254	10.47.255.255
<b>10.48.0.0</b>	10.48.0.1- 10.63.255.254	10.63.255.255
<b>10.64.0.0</b>	10.64.0.1- 10.207.255.254	10.207.255.255
<b>10.208.0.0</b>	10.208.0.1- 10.223.255.254	10.223.255.255
<b>10.224.0.0</b>	10.224.0.1- 10.239.255.254	10.239.255.255
<b>10.240.0.0</b>	10.240.0.1- 10.255.255.254	10.255.255.255

1. The address 10.212.10.50/12 is a host address in the 10.208.0.0/12 subnet.
2. The network address for the subnet is 10.208.0.0 and the broadcast address is 10.223.255.255

## 2-4 Variable Length Subnet Masks (VLSM)

VLSM and our next topic, summarization builds up on subnetting. If you still have doubts on subnetting, I would strongly suggest you devote some more time to it and practice before moving ahead. You also may want to consider picking up our 100 page How & Why We Subnet Workbook. This workbook walks you through over 60 examples to help you really understand the ins and outs of subnetting.

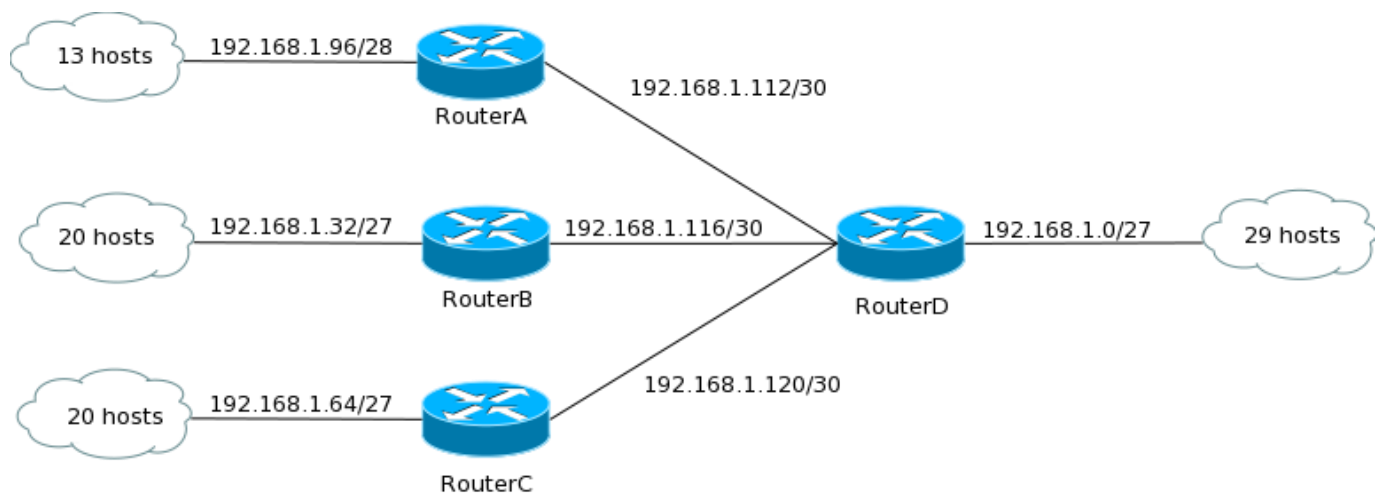
Figure 2-1 *Classful Network*



Earlier, it was required to use the same subnet mask across the network. This was called **classful networking**. With increase in complexity of networks and decrease in available IP addresses it became obvious that classful networking causes waste valuable of IP addresses. To understand how, consider Figure 2-1. The largest subnet requires 30 host addresses. So across the network a mask of /27 is used, which gives 30 hosts per subnet. You will notice that in every subnet except the subnet attached to RouterD, some host addresses will remain unused. In particular, 28 host addresses are wasted for each link between the routers. In total this network wastes 118 addresses and uses 92 addresses.

To avoid wasting of IP addresses, **classless networking** was introduced by way of VLSM. VLSM allows you to use different subnet masks across the network for the same class of addresses. For example, a /30 subnet mask, which gives 2 host addresses per subnet, can be used for point-to-point links between routers. Figure 2-2 shows how VLSM can be used to save address space in the network shown in Figure 2-1.

Figure 2-2 *Classless Network with VLSM*



In Figure 2-2, notice the different masks used for each subnet. The first network with 13 hosts is using a mask of /28, which gives 16 hosts addresses. The point-to-point links between the routers are using a /30 mask which gives 2 host addresses. In total the network is still using 92 addresses but is wasting only 22 addresses. Now that you know the benefit of VLSM, take a look at how you can use it in a network.

There are a few restrictions you need to consider when planning to use VLSM:

1. You need to use routing protocols that support classless routing such as Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), Border Gateway Protocol (BGP) or Routing Information Protocol (RIP) version 2. Classful protocols such as RIPv1 cannot be used with VLSM. While routing protocols are covered in detail in Chapter 4, you should understand that a routing protocol is classful because it does not advertise the subnet mask along with the network address in its updates. Hence, routers running these protocols, do not know the subnet mask and strictly follow the class of the network. Classless protocols on the other hand advertise and understand subnet masks.
2. You need to use fixed block sizes. You have come across these block sizes during subnetting practice and these are listed in Table 2-9. You cannot use any block sizes apart from these. For example in Figure 2-2, for the networks connected to RouterB and RouterC, a block size of 32 was used even though the total addresses required were 21 in each subnet.

**Table 2-9** Block Sizes for VLSM

Block Size	Host addresses available
128	126
64	62



<b>32</b>	30
<b>16</b>	14
<b>8</b>	6
<b>4</b>	2

When designing a network using VLSM, the following simple steps can help come up with an appropriate addressing scheme:

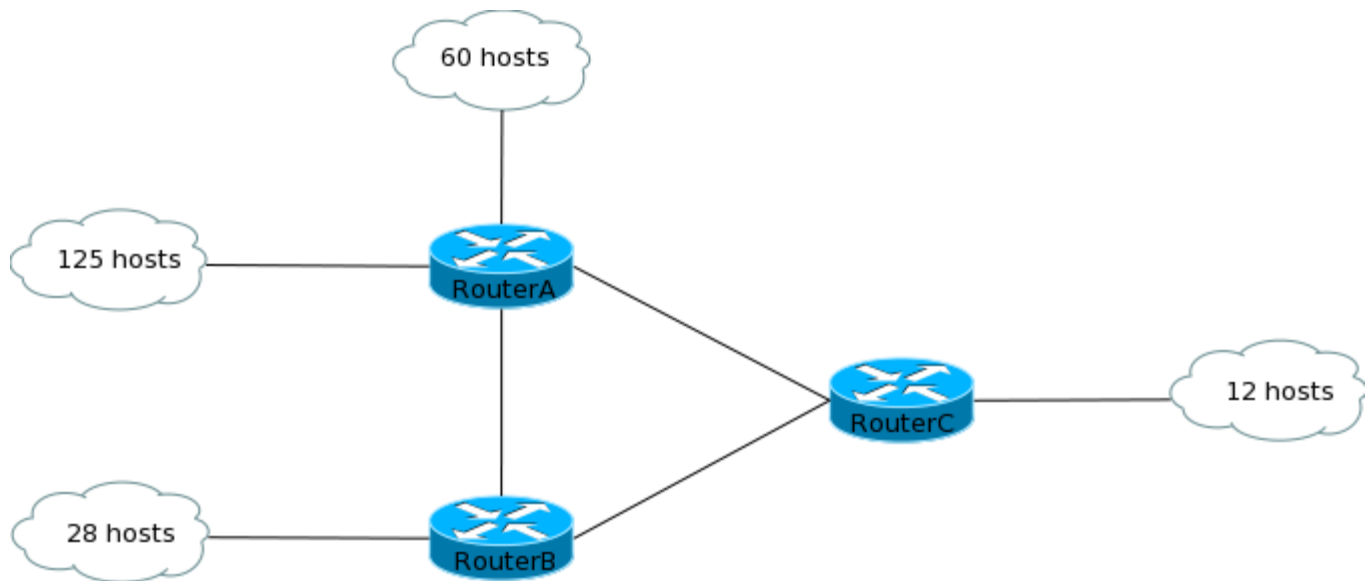
1. Start by finding the largest subnet in your network. The number of host addresses needed decides the size of the subnet.
2. Next assigning an appropriate mask to the largest subnet using the block sizes mentioned in Table 2-9.
3. Note the subnet numbers remaining with the mask used in Step 2.
4. Take the next available subnet and subnet it further to accommodate your smaller subnets.
5. Write down your new subnet numbers again.
6. Repeat step 4 and 5 for smaller segments.

Consider the example shown in Figure 2-2 and work through the above steps to see how the network address and subnet mask was found for each segment:

1. The largest segment in Figure 2-2 is attached to RouterD. It requires 30 host addresses, including the router interface (29 host addresses and 1 router interface). So we can use a /27 mask which gives us exactly 30 host addresses. We assign 192.168.1.0/27 to that subnet.
2. Our new subnets using /27 mask are 192.168.1.0/27, 192.168.1.32/27, 192.168.1.64/27, 192.168.1.96/27, 192.168.1.128/27 etc.
3. Next we look at the smaller subnets. The subnets attached to RouterB and RouterC require 21 host addresses (20 host addresses and 1 router interface). The block size we can use for them is 32. We already have subnets available with /27 mask, so we simply assign them to these segments – 192.168.1.32/27 and 192.168.64/27.
4. Our next smaller segment is the one attached to RouterA. It requires 14 host address, so a block size of 16 or a mask of /28 can be used. So we take the next available subnet, 192.168.1.96/27 and subnet it further using a /28 mask. This gives us 192.168.1.96/28 and 192.168.1.112/28. We assign the first of these to this segment – 192.168.1.96/28.
5. Finally we have the three point-to-point segments between the routers. Each requires 2 host addresses hence a block size of 4 and a mask of /30. We take our available subnet – 192.168.1.112/28 and subnet it further using a mask of /30. This gives us 192.168.1.112/30, 192.168.1.116/30, 192.168.1.120/30 and 192.168.1.124/30. We use the first three for these segments – 192.168.1.112/30, 192.168.1.116/30 and 192.168.1.120/30.

Consider Figure 2-3 as another example. Using a class C network of 192.168.10.0/24 design a VLSM solution to accommodate host requirements of all the segments.

**Figure 2-3** VLSM – Example #2



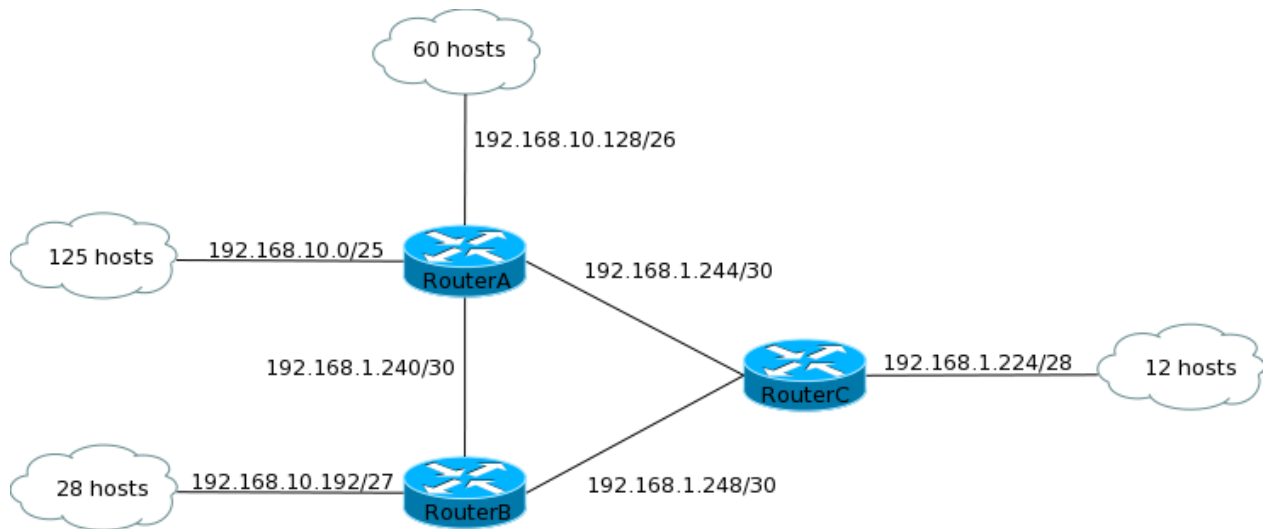
To design the VLSM solution, follow the 5 steps discussed earlier:

1. The largest segment requires 125 host addresses. So a mask of /25 can be used. This gives two subnets – 192.168.10.0/25 and 192.168.10.128/25. The first subnet can be assigned to this segment.
2. The second largest segment requires 60 host addresses. You can take the second available subnet – 192.168.10.128/25 – and divide it further using a /26 mask to give you subnets 192.168.10.128/26 and 192.168.10.192/26. Assign the first one to this segment.
3. The third largest segment requires 29 host addresses (28 host addresses and 1 for the router interface). You will need to use a block of 32 and a mask of /27. Take the remaining subnet from the previous step and divide it further using a /27 mask. This will give you subnets 192.168.1.192/27 and 192.168.1.224/27. Assign the first one to this segment.
4. The fourth largest block requires 13 host addresses (add one for the router interface). You can use a block of 16 and a mask of /28. Take the remaining subnet from the previous step and divide it further using a mask of /28. This will give you subnets 192.168.1.224/28 and 192.168.1.240/28. Assign the first one to this segment.
5. Now you are left with 3 point-to-point links between the routers. These links require two host addresses and a mask of /30. Take the remaining subnet from the previous step and divide it using a mask of /30. This will give you subnets 192.168.1.240/30, 192.168.1.244/30, 192.168.1.248/30 and 192.168.1.252/30. Use the first three of these for the point-to-point links. The remaining one subnet can be left for future use.

## TUN MIN OO {BE-IT} Routing & Switching 200-120

Figure 2-4 shows the solution derived in the above steps.

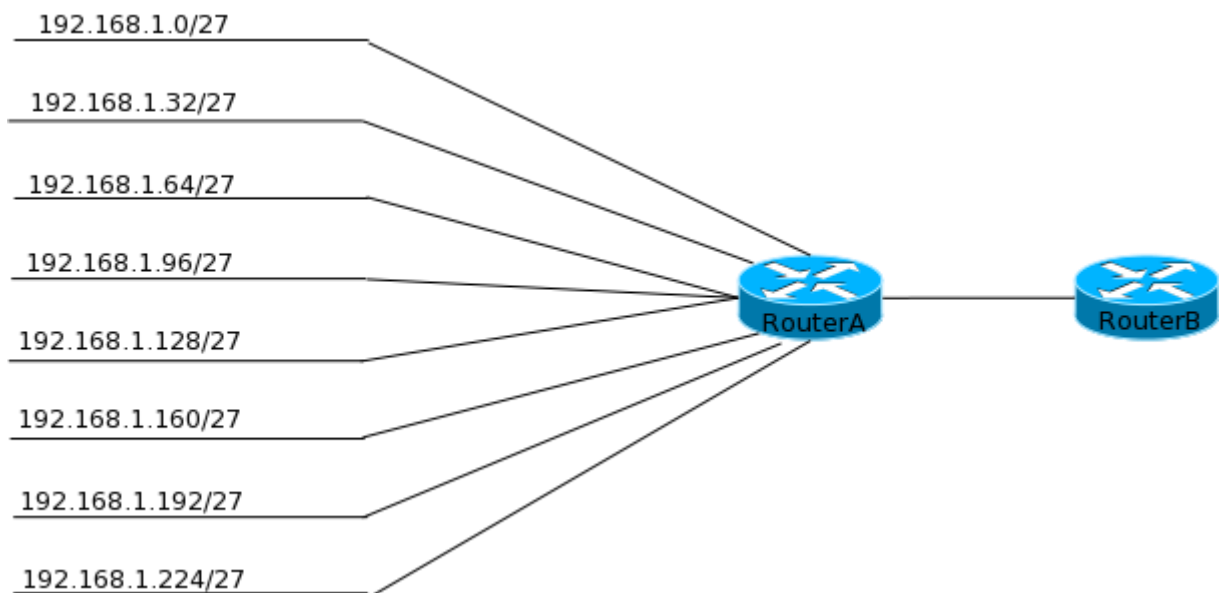
**Figure 2-4** VLSM – Solution for Example #2



## 2-5 Route Summarization

You already know from the previous chapter that routers function by creating a table of all networks it knows about. This table is called the routing table and routers use routing protocols to tell each other about the networks they know of. As networks increase, so do the number of entries in a routing table. Large routing tables cause increased processing and lower response time in a router. To reduce the size of routing tables, networks can be grouped together or **summarized** using a mask that incorporates them all. For example, in figure 2-5, a 192.168.10.0/24 subnet has been divided into smaller subnets of /27 mask. All of these networks connect to RouterA which in turn is advertising these routes to RouterB. Without summarization, RouterB will come to know of 8 networks which are available via RouterA. Since these networks are contiguous subnets that can have been subnetted from a /24 address, they can be summarized back into 192.168.1.0/24 network by RouterA while advertising to RouterB. This way, RouterB comes to know of one large /24 network only instead of 8 smaller /27 networks.

Figure 2-5 Summarization



Summarization is similar to VLSM but in the opposite direction. When using VLSM you move to the right in terms of the bits (/24 to /25, /25 to /26, so on and so forth) while during summarization you move to the left (example /27 to /24).

Summarization is somewhat simple if you remember the following:

1. You can only summarize in the block sizes you learned about in VLSM – 128,64,32,16,8,4.
2. The network address used for the summarized address is the first network address in the block.

## TUN MIN OO {BE-IT} Routing & Switching 200-120

For example, if you want to summarize networks 192.168.8.0 through 192.168.15.0, first find the block size you can use. There are 8 networks so the block size of 8 can be used. The first network address in the block is 192.168.8.0. Now to find the mask of the summarized route, remember the mask used for a block of 8 – 248. You can also deduct the block size from 256 to find the mask. Since we are summarizing the third octet the subnet mask for the summary address will be 255.255.248.0.

Take another example, 172.16.0.0 through 172.16.35.0. This one is not as simple as the first one. Notice that you have 36 networks to summarize which does not conform to the block sizes. There are two things that you can do here:

1. Summarize in block size of 32 (mask of 224). This will give you a summary address of 172.16.0.0 255.255.224.0 but will only summarize networks 172.16.0.0 through 172.16.31.0. The rest of the 4 networks will be advertised as individual routes.
2. Summarize in block of 64 (mask of 192). This will give you a summary address of 172.16.0.0 255.255.192.0 but will summarize networks 172.16.0.0 through 172.16.63.0.

The correct answer depends on the network. If you are planning to add networks 36 to 63 then the second options works. Otherwise the first option is the best one.

Take a third example where you know the summary address of 172.10.16.0 with a mask of 255.255.224.0 and need to find which networks are being summarized. This is really easy. The third octet is the interesting octet and gives a block size of 32. This means the networks 172.10.16.0 through 172.10.47.0 have been summarized.

As a final example, consider the following networks:

- 192.168.1.0/25
- 192.168.1.128/25
- 192.168.2.0/24
- 192.168.3.0/24
- 192.168.4.0/26
- 192.168.4.64/26
- 192.168.4.128/26
- 192.168.4.192/26

Try to figure out the summary address that can be used for these networks. If you look carefully the third octet forms a contiguous block of 4 and can be summarized with the address 192.168.1.0 255.255.252.0 or 192.168.1.0/22.

In the last example notice that we summarized a contiguous block of class C using a mask. This is called **supernetting**. Supernetting is an extension of VLSM and summarization. In summarization you summarize networks subnetted while in supernetting you summarize a block of contiguous blocks of

## TUN MIN OO {BE-IT} Routing & Switching 200-120

Class A, B or C networks. Supernetting is usually practiced by ISPs to reduce the Internet routing table size.

## **2-6 Troubleshooting IP Addressing**

As you know by now, IP Addressing is an integral part of networking and given the complexity of addressing and subnetting, it is common to have IP addressing errors in the network. So it is essential for you to be able to troubleshoot common problems related to IP Addressing. Before troubleshooting a network, you have to understand the below given common protocols and utilities that are used to troubleshoot:

- **Packet InterNet Grouper (PING)** – Ping is one of the most commonly used utility that is used to troubleshoot addressing and connectivity problems. This utility is available in almost all operating systems, including Cisco devices and can be accessed by the command line interface using the **ping** command. It uses the ICMP protocol to check if the destination host is live or not.
- **Traceroute** – Traceroute is another common utility that is available with all operating systems. In some operating systems the utility can be access using the **tracert** or **traceroute** command on the CLI. It is used to find each hop between the source and destination hosts and is useful to see the path taken by a packet.
- **ARP table** – Sometimes it is useful to look at the ARP table of a system. This table contains the MAC address to IP address bindings learned by the system. On most operating systems the ARP table can be viewed using the **arp -a** command. On a Cisco device the arp table can be viewed using the **show ip arp** command.
- **IP config** – Sometimes, you need to verify the IP address, subnet mask, default gateway and DNS addresses the host is using. On a windows machine all this information can be seen in the output of the **ipconfig /all** command. On a unix based system, this information can be seen using the **ifconfig** command.

For the following section consider the network shown in Figure 2-6. In this network, HostA is trying to reach ServerA and ServerB but is not able to.

Before looking at the IP addressing, you should quickly check network connectivity using four steps that Cisco recommends:

1. Ping 127.0.0.1, the loopback address from the Host. You will need to open a terminal window of your operating system to use the ping utility. If you get an output similar to the following, it shows that the IP stack in the host is working well:

```
ping 127.0.0.1
```

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
```

```
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.073 ms
```

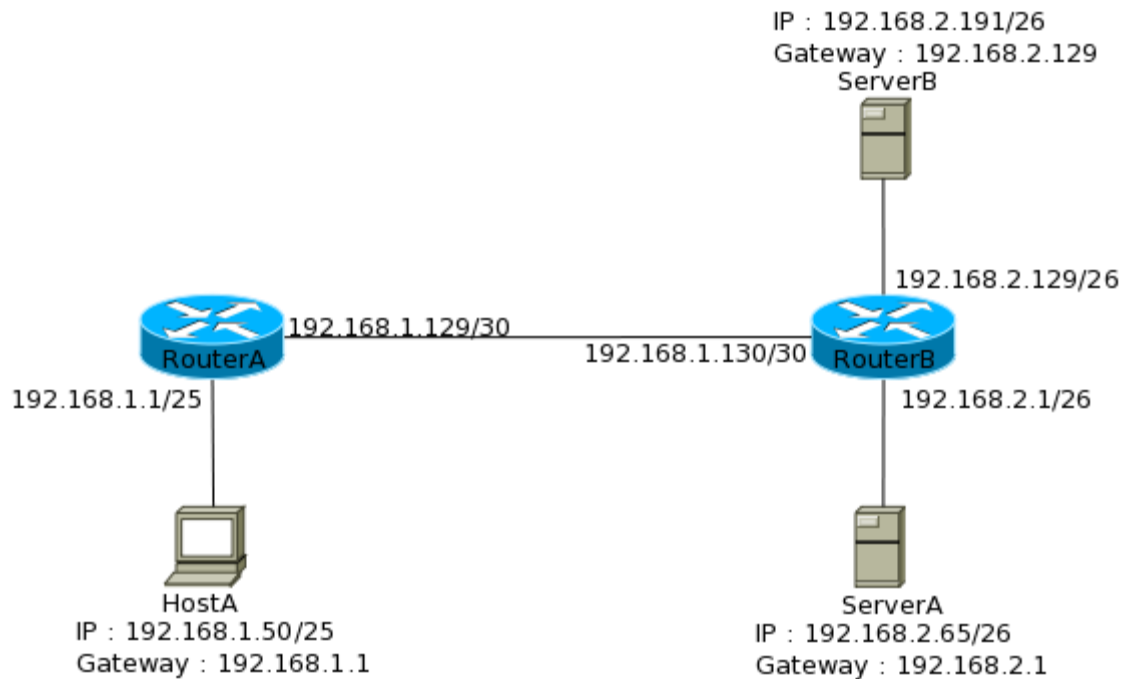
## TUN MIN OO {BE-IT} Routing & Switching 200-120

64 bytes from 127.0.0.1: icmp\_seq=1 ttl=64 time=0.096 ms

64 bytes from 127.0.0.1: icmp\_seq=2 ttl=64 time=0.095 ms

64 bytes from 127.0.0.1: icmp\_seq=3 ttl=64 time=0.145 ms

Figure 2-6 Troubleshooting IP Addressing Scenario



2. Ping the IP address of the host itself. If its successful then it shows that the host's NIC is working well.

```
>ping 192.168.1.50
```

```
PING 192.168.1.50 (192.168.1.50): 56 data bytes
```

64 bytes from 192.168.1.50: icmp\_seq=0 ttl=64 time=0.075 ms

64 bytes from 192.168.1.50: icmp\_seq=1 ttl=64 time=0.096 ms

64 bytes from 192.168.1.50: icmp\_seq=2 ttl=64 time=0.155 ms

64 bytes from 192.168.1.50: icmp\_seq=3 ttl=64 time=0.151 ms

3. Ping the default gateway from the host. If the ping works it shows that your host is able to communicate with the network and the default gateway.



```
>ping 192.168.1.1
```

```
PING 192.168.1.1 (192.168.1.1): 56 data bytes
```

```
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=0.075 ms
```

```
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.096 ms
```

```
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.155 ms
```

```
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.151 ms
```

4. Finally ping the remote host, ServerA or ServerB in our case. If the ping is successful, this means there is a DNS or application layer protocol problem between the host and ServerA. However, in our case the ping fails.

```
>ping 192.168.2.65
```

```
PING 192.168.2.65 (192.168.2.65): 56 data bytes
```

```
Request timeout for icmp_seq 0
```

```
Request timeout for icmp_seq 1
```

```
Request timeout for icmp_seq 2
```

```
Request timeout for icmp_seq 3
```

Now that you have used the Cisco recommended way to determine that the problem lies in the network, it is time to look at the addressing. In this exercise, you need to look at the IP address, subnet mask and default gateway configured (as shown in Figure 2-6) to see if they are correctly configured. You can simply look at the subnet mask and see which are valid host addresses in that subnet to see if valid IP addresses have been configured. Take a step-by-step approach as shown below to narrow down the problem area:

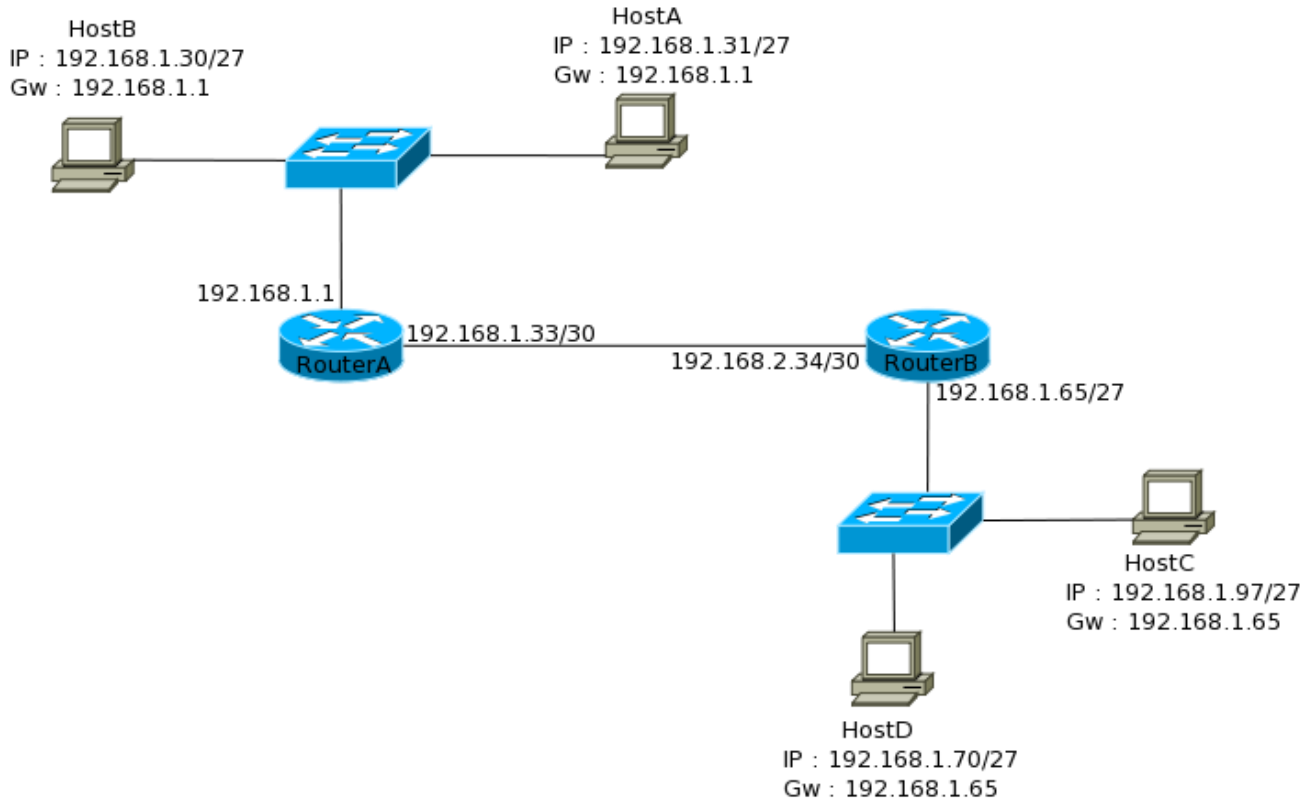
1. The Host has an IP address of 192.168.1.50/25. A mask of /25 shows that the host lies in the 192.168.1.0/25 subnet (/25 = 255.255.255.128, which gives two subnets – 0 and 128). So the IP address given to the host is a valid host address.
2. The Gateway address on the host is 192.168.1.1 and that is the IP address on the Router interface connected to the network. The IP address lies in the same subnet range as the host address. Step 1 and Step 2 eliminate addressing problem in the network segment to which the host is connected.

3. The next network segment is the point-to-point link between RouterA and RouterB. The subnet mask of /30 gives subnets 0,4,8,12....128. The valid host addresses in the network 192.168.1.128/30 are 192.168.1.129 and 192.168.1.130. So the point-to-point links have valid addresses.
4. The next network segment is the one to which ServerA is connected. /26 mask converts to 255.255.255.192. 192 deducted from 256 leaves 64. This means the valid subnets are 192.168.2.0, 192.168.2.64, 192.168.2.128, 192.168.2.192. ServerA's address is a valid address in the 192.168.2.64 subnet but the default gateway and the router's address is in the 192.168.2.0 subnet. So ServerA's address is in the wrong subnet and needs to be changed to a valid address in the 192.168.2.0 subnet. This explains why HostA is not able to reach ServerA.
5. The final segment is the one to which ServerB connects. From the calculations done in the previous step, you can see that ServerB's address lies in the 192.168.2.128 subnet. The valid host addresses in this subnet are 129 to 190. 191 is the broadcast address of the subnet. While the router (default gateway) is configured with a valid address, ServerB has been assigned the broadcast address, which needs to be changed. This explains why HostA is not able to reach ServerB.

If you are careful about going step-by-step and finding out valid addresses in each subnet, you can figure out any addressing problem in no time. Lets take a look at another example two examples. For these examples, we will use the network shown in Figure 2-7.

**Figure 2-7 Troubleshooting IP Address – Example #2 & #3**

## TUN MIN OO {BE-IT} Routing & Switching 200-120



### Example #2

Problem: HostB is able to reach HostD but it is not able to reach HostA

Solution: The question tells us two things. First that HostB is able to reach HostD, that means the network from HostB all the way to HostD is working fine. Second, HostB is not able to reach HostA. It is simple to figure out that there is a problem at HostA. To find the problem, take a look at the IP address information given for HostA:

1. A subnet mask of /27 converts to 255.255.255.224.
2. Deducting 224 from 256 gives us 32. So the valid host subnets are 0, 32, 64 and so on.
3. HostB and RouterA's address are in the 192.168.1.0/27 subnet that has a valid host range of 1 to 30. The broadcast address for this subnet is 192.168.1.31.
4. You will notice that HostA has an IP address of 192.168.1.31/27, which is the broadcast address of this subnet and not a valid host address. Hence, HostA cannot be reached from the network.

### Example #3

Problem: HostD is able to reach HostB but not HostC.

Solution: Again this problem statement tells us that the network from HostD to HostB is working well. So the problem requires a look at HostC's addressing:

1. Again, a mask of /27 gives us subnets 0, 32, 64, 96, 128 and so on.
2. HostD and RouterB's addresses lie in the 192.168.1.64/27 network. The valid host addresses for this subnet are 192.168.1.65-94. The broadcast address for the subnet is 192.168.1.95.
3. The next subnet is 192.168.1.96/27 that has a valid host range of 192.168.1.97-192.168.1.127.
4. You will notice that the IP address of HostC lies in the 192.168.1.96/27 subnet and not the 192.168.1.64/27 subnet. It lies in a different subnet than the default gateway (RouterB) and HostD. Hence, HostD is not able to reach HostC.



**Exam Alert:** Expect a lot of questions in different forms where such IP addressing errors will be hidden during the exam. Each time you will need to patiently find the subnet and valid host addresses.

### Broadcast Addresses

Broadcast and broadcast addresses are discussed many times in Chapter 1 and Chapter 2. Broadcast is a generic term meaning message or data sent to all hosts in a network while broadcast address is a generic term meaning an address to which broadcasts are sent. It is important to understand that not all broadcasts are same. They can be divided into two different types:

- **Layer 2 broadcasts** – These broadcasts are sent at layer 2 and are limited to a LAN. These do not cross the boundary of a LAN, which is defined by a router.
- **Layer 3 broadcasts** – These broadcasts are sent at layer 3 and go to the network.

You already know what **unicast** and **multicast** are but just to put them into perspective of broadcasts, these terms are defined below again:

- **Unicast** – Messages or data sent to a single host are called unicast.
- **Multicast** – Messages or data sent to a group of devices is called multicast.

Like broadcasts, broadcast addresses also differ based on the layer. The different types are discussed below:

- **Layer 2 Broadcast Address** – Layer 2 addresses are 48bit hexadecimal values. An example of layer 2 addresses is a3.4c.56.ea.f5.aa. Similarly, a layer 2 broadcast is a hexadecimal value of all Fs or a binary value of all 1s – FF.FF.FF.FF.FF.FF
- **Layer 3 Broadcast Address** – This chapter showed you that the last address of a subnet is a broadcast address such as 192.168.1.255/24. These addresses have all host bits on and refer to all hosts in that subnet. An address with all its bits turned on – 255.255.255.255 – is a special broadcast address that refers to all hosts in all networks.

## TUN MIN OO {BE-IT} Routing & Switching 200-120

A good example to understand how broadcast addresses are used, consider the following example of how a host requests IP address from a DHCP server:

- When a host boots up and needs to get an IP address from the DHCP server, it does not know if the DHCP server is in this same LAN segment or across a router. So it sends a DHCP request with the destination IP address set to 255.255.255.255 and the destination MAC address set to FF.FF.FF.FF.FF.FF
- The layer 2 broadcast goes out to the LAN and if a DHCP server is connected to the segment, it will respond back.
- If the DHCP server is not on the segment, the router will see the packet and convert it into a unicast message and send it to the DHCP server. The router needs to be configured for this though.
- The DHCP will reply back with a unicast.

As the above example demonstrates, broadcast is very useful and can be converted to unicast when required.

### Summary

This chapter is one of the most important chapters in this book and covers the most fundamental blocks of a network. IP Address Classes, Private and Public addresses and subnetting are very important for both the CCNA exam as well as for understanding the rest of the topics coming up

I cannot stress enough the importance of these topics and would strongly suggest you to go through it again and clarify any doubts you might have before moving ahead.

## **Chapter 3**

### ***Introduction to Cisco Routers, Switches and IOS***

#### **3-1 Introduction to Cisco Routers, Switches, IOS & the Boot Process**

The previous two chapters helped you learn the basics of networking. You are aware of various layers of the OSI and TCP/IP models and the devices that work on these layers, especially routers and switches. The rest of the book focuses on various functions of Cisco routers and switches. So before moving to the various functions, it is necessary to know what makes them tick. This chapter is dedicated to **Cisco Internetwork Operating System (IOS)**. Cisco IOS is a proprietary operating system that Cisco routers and switches run on. This chapter looks at the boot process, connectivity options, ways to configure the devices and show basic configuration and verification commands.

#### **Cisco Integrated Services Router (ISR)**

Cisco provides various series and models of routers geared towards different types of customer and requirements. Some of them just do routing whereas others provide some other functions such as Wireless connectivity, Security features and Voice-over-IP services. Cisco's ISR series routers are example of routers that provide various services.

The earlier CCNA exams used to focus on Cisco 2500 and 2600 routers that have been replaced by ISR 1800 and 2800/2900 series routers. 2500 and 2600 routers are End-of-Life now and cannot be bought from Cisco anymore. Figure 3-1 shows a part of the backplane of a Cisco 1841 router with important parts labeled. These parts are described in Table 3-1. Figure 3-2 shows the front panel of the router.



**Exam Alert:** CCNA is not a device specific exam. You can practice using a 2500 or 2600 router or even a 3800 series ISR router. Every command and concept discussed in this book holds true for all of these routers. The only difference that you need to be aware of is the output difference in memory, interface type (Ethernet or FastEthernet) and number of interfaces

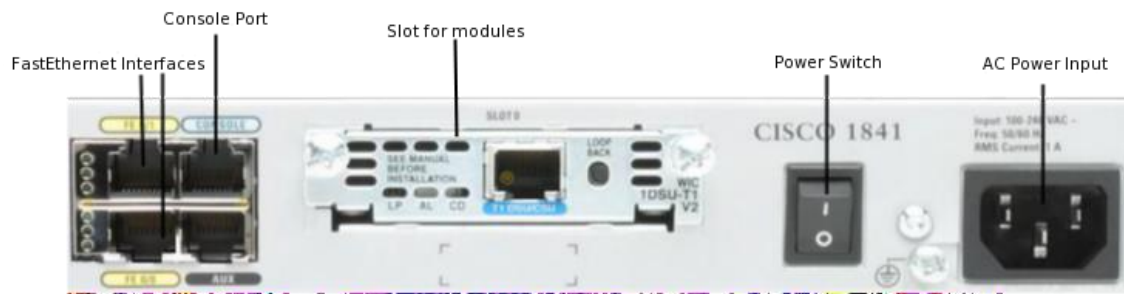


Figure 3-1 Rear view of a Cisco1800 Series ISR



Figure 3-2 Front of a Cisco1800 Series ISR

Backplane component	Description and Usage
<b>FastEthernet Interfaces</b>	These are FastEthernet interfaces used to connect the router to the network. Different routers have different number of interfaces. Most of them have slots which allows you to add a module containing more interfaces. Apart from Fastethernet interfaces, a router can have serial interfaces (for WAN connection), an ADSL interface and many other interfaces. Some of these are discussed later in the book while most of beyond the scope of CCNA.
<b>Console Port</b>	This port used to connect to the router to configure, monitor and troubleshoot. More on connecting to the router is discussed shortly.
<b>Slot for Modules</b>	Some routers have slots where additional modules can be added. These modules usually add interfaces to the router.
<b>Power Switch</b>	To switch on or off the router
<b>AC Power Input</b>	To provide power supply to the router.

Table 3-1 Rear components of the router

### Cisco Catalyst Switches

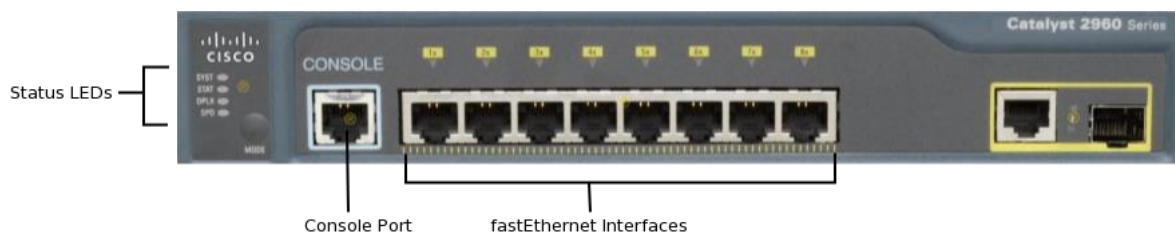
Cisco provides a wide range of switches under its Catalyst brand. The Catalyst brand encompasses many series of switches with each series targeting a particular part or size of a network. The CCNA exam focuses on the 2960 series of switches in the Catalyst brand. 2960 switches are low-cost wiring closet switches that you would expect to be used at the Access layer (remember the Cisco Hierarchical model) for providing network connectivity to hosts.



**Exam Alert:** As with routers, you can use any switch model as long as it runs IOS when studying for your CCNA exam. I suggest practicing with either a 2950 or a 2960 switch. If your budget can afford one, a 3550 or 3560 Layer 3 switch can be used with its enhancements. But stay away from the 4000 or 6000 series switches.

Each model in the 2960 series switch is different in terms of the number of physical network interfaces it has but overall each model looks similar. Figure 3-3 shows the front faceplate of the switch. The back of the switch only consists of the AC power input.

Table 3-2 describes the important components shown in Figure 3-3.



**Figure 3-3** *Front plane of a Cisco Catalyst 2960 Switch*



Backplane component	Description and Usage
<b>FastEthernet Interfaces</b>	These are FastEthernet interfaces used to connect the hosts to the network. Different models have different number of interfaces. Some high end switches can have hundreds of these interfaces.
<b>Console Port</b>	It is a port used to connect to the switch to configure, monitor and troubleshoot. More on connecting to the switch is discussed shortly.
<b>Status LEDs</b>	These LEDs show the status of various components of the switch. Apart from these, there is a LED over each interface showing the status of that interface. Each LED can be either off, amber or green.

**Table 3-2 Backplane components of a router**

## Cisco Internetwork Operating System (IOS)

Cisco IOS (different from Apple's iOS) is a proprietary kernel which controls all functions of a Cisco router and most switches. Cisco IOS is based on the operating system created by William Yeager at Stanford University between 1980 and 1986. Cisco licensed Yeager's work and created the IOS out of it. The Cisco kernel allocates resources and manages things such as low-level hardware interfaces and security.

Some important items that the Cisco router IOS is responsible for include:

- Carrying network protocols and functions
- Connecting high-speed traffic between devices
- Adding security to control access and stop unauthorized network use
- Providing scalability for ease of network growth and redundancy
- Supplying network reliability for connecting to network resources

Apart from the routing, switching, telecommunications and security functions, the IOS also provides a **Command Line Interface (CLI)** for configuration, management, monitoring and troubleshooting. The CLI can be access using the console port, the auxiliary port (if it is available) and Telnet or SSH. Telnet or SSH access requires IP connectivity, hence the initial configuration requires you to access the device using the console port.

The rest of the chapter is dedicated to connecting to the CLI and basic configuration.

## Connecting to the CLI using Console port

To get to the CLI of Cisco router or switch you will need to connect your PC to the console port of the device. The console port on a Cisco router or switch is a RJ45 port. You need to use a UTP rollover cable (discussed in Chapter 1) with RJ45 connector on one end to insert into the router or switch's console port and there will be a 9 pin serial connection on the other end which you will plug into a 9 pin serial port on your computer. Cisco ships a blue console cable with almost every device. \*Note: Many computers today do not come with a 9 pin serial port so you will need to purchase a 9 pin serial to USB converter and put this on the end of your Cisco console kit so you can make the physical connection.

Connect the serial connector end to the serial port of your PC and the RJ45 connector to the console port of the router or switch. After the physical connection, you will need to use software known as a Terminal Emulator to connect to the CLI. HyperTerminal is an example of a Terminal Emulator that comes pre-installed on some Windows systems. If you do not have HyperTerminal on your Windows PC, you may want to download PuTTY which is a free terminal emulator. Minicom is a free terminal emulator for Unix based operating systems.

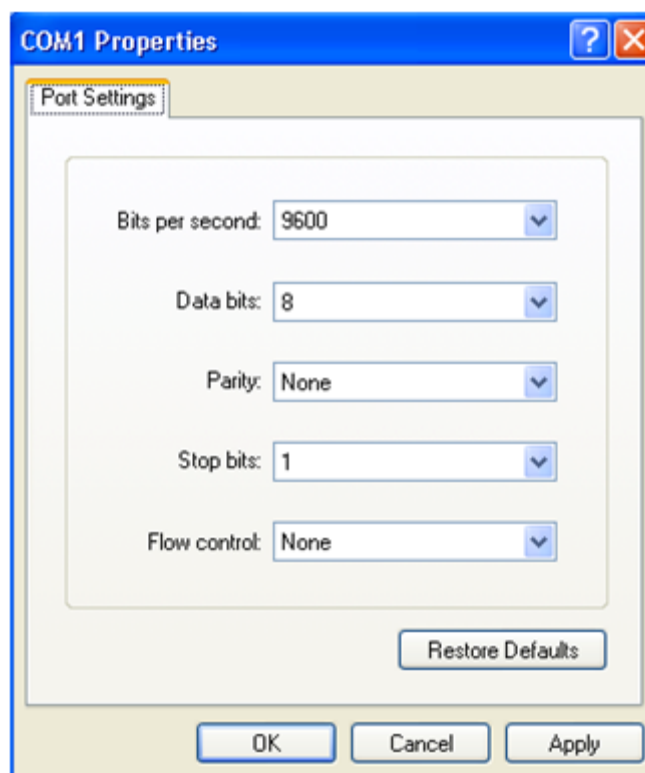


Figure 3-4 Hyperterminal configuration to connect to IOS CLI

## TUN MIN OO {BE-IT} Routing & Switching 200-120

Launch your terminal emulator and configure it to connect to the serial interface using the following settings:

- 9600 bits/second
- 8 data bits
- Parity None
- 1 stop bit
- No flow control

Figure 3-4 shows Hyperterminal configured to use the above settings.

### Booting Up a Router or a Switch

When you power up a Cisco router or a switch, it first runs the Power-On Self-Test (POST). After POST the device looks for and loads the Cisco IOS from flash memory. Flash memory is an Electronically Erasable Programmable Read-Only Memory (EEPROM). When the IOS loads, it looks for the configuration file in the non-volatile RAM or NVRAM. Take a look at the booting process of a Cisco Router shown below. The following output is from an 1841 router.

```
System Bootstrap, Version 12.3(8r)T8, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2004 by cisco Systems, Inc.
PLD version 0x10
GIO ASIC version 0x127
c1841 processor with 131072 Kbytes of main memory
Main memory is configured to 64 bit mode with parity disabled
```

The first part above shows information regarding the bootstrap program that runs the POST and then tells the router to load IOS. By default the location of the IOS is the flash memory. The next part shows the IOS image being decompressed. The pound sign shows the progress of the decompression process.

[output truncated]

```
Self decompressing the image : ##### [OK]
```

## TUN MIN OO {BE-IT} Routing & Switching 200-120

After decompression, the IOS is loaded in the RAM and starts to run. During the startup a lot of information is shown. In the output below notice the IOS version shown as 12.4(25e).

*[output truncated]*

*Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M), Version 12.4(25e), RELEASE SOFTWARE (fc2)*

*Technical Support: <http://www.cisco.com/techsupport>*

*Copyright (c) 1986-2011 by Cisco Systems, Inc.*

*Compiled Wed 16-Mar-11 14:42 by prod\_rel\_team*

Once IOS has completely loaded, it will display important information about the router that was learned during POST and booting. You can see the make of the router, the flash and RAM size as well as various interfaces and modules connected as you can see below:

*[output truncated]*

*Cisco 1841 (revision 5.0) with 115712K/15360K bytes of memory.*

*Processor board ID FRT072 158RA*

*2 FastEthernet interfaces*

*1 Virtual Private Network (VPN) Module*

*DRAM configuration is 64 bits wide with parity disabled.*

*191K bytes of NVRAM.*

*31360K bytes of ATA CompactFlash (Read/Write)*

The output above shows that the 1841 Cisco router has a total memory of 128MB RAM, 191KB of NVRAM and 32MB of Flash. It also has 2 FastEthernet Interfaces.

Once IOS is loaded, it will copy the saved configuration, called the startup config, from the NVRAM into the RAM. This copy is known as the running config.



**Exam Alert:** The boot up sequence and type of messages will be similar across all routers. The only noticeable differences will be the reported size of RAM, NVRAM, flash and the number/type of interfaces. Expect to see this type of output in your CCNA simulation exam questions.

## TUN MIN OO {BE-IT} Routing & Switching 200-120

The boot process of a Cisco catalyst switch is similar. The following outputs show the messages that appear when a 2950 switch is booted up.

```
C2950 Boot Loader (C2950-HBOOT-M) Version 12.1(11r)EA1, RELEASE SOFTWARE (fc1)
Compiled Mon 22-Jul-02 17:18 by antonino
WS-C2950G-24-EI starting...
```

The above message shows the bootstrap program running. The output below shows the IOS being decompressed and then loaded into the RAM.

[output truncated]

```
Loading "flash:/c2950-i6q4l2-mz.121-22.EA6.bin"...#####
```

```
File "flash:/c2950-i6q4l2-mz.121-22.EA6.bin" uncompressed and installed, entry point: 0x80010000
```

```
executing...
```

After the IOS is decompressed, the IOS version is displayed. Note the version displayed below is 12.1(22)EA6.

[output truncated]

```
Cisco Internetwork Operating System Software
IOS™ C2950 Software (C2950-I6Q4L2-M), Version 12.1(22)EA6, RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Fri 21-Oct-05 01:59 by yenanh
```

After IOS loads, it runs POST on various components of the switch as can be seen below.

[output truncated]

```
POST: System Board Test : Passed
POST: Ethernet Controller Test : Passed
ASIC Initialization Passed
POST: FRONT-END LOOPBACK TEST : Passed
```

After the last POST is passed IOS completes loading and displays the information learned during the POST. The output is similar to the one displayed when the router completes booting and provides information regarding the device.

```

cisco WS-C2950G-24-EI (RC32300) processor (revision L0) with 21013K bytes of memory.
Processor board ID FOC1028Y1TA
Last reset from system-reset
Running Enhanced Image
24 FastEthernet/IEEE 802.3 interface(s)
2 Gigabit Ethernet/IEEE 802.3 interface(s)
32K bytes of flash-simulated non-volatile configuration memory.
[output truncated]

```

The output above shows that the 2950 switch has 20MB of RAM and 32KB of flash. There are 24 FastEthernet interfaces and 2 Gigabit Ethernet Interfaces in the switch. Just as in the case of the Router, once IOS has loaded, it will copy the startup config into the RAM as running config.

In both, the case of the router as well as the switch, if startup config is not present, the device will go into the **setup mode** and start the System Configuration dialog. This is a step-by-step process to help you with basic configuration. You can tell that the device has gone into the setup mode if you see the following output after IOS loads:

```

— System Configuration Dialog —
Would you like to enter the initial configuration dialog? [yes/no]:
% Please answer 'yes' or 'no'.

```

You will not be going through the setup mode since CCNA is all about configuring the switches and the routers using the CLI.

Table 3-3 sums up all the components and their functions that you learned about in this section.

Component	Function
<b>Bootstrap</b>	A small program that runs the POST test and then loads the IOS on bootup.
<b>Flash Memory</b>	An EEPROM where the IOS file is stored. The bootstrap looks for the IOS file here first.
<b>RAM</b>	The working memory of the device. A copy of the configuration is also stored here after bootup.
<b>NVRAM</b>	Non-volatile RAM it stores a copy of the configuration. On bootup, IOS reads the configuration file from here.

**Figure 3-3 Important components used during boot**



**Note:** The rest of the chapter is dedicated to basic configuration using the CLI and the commands and concepts apply to both a router and a switch, unless specifically mentioned otherwise. The CCNA exam only uses the CLI and no GUI at this time.

—————*GO TO 3-2*—————

### **3-2 Using the Command-Line Interface (CLI)**

Once IOS has finished loading up, it will ask you to press Return to continue. While waiting for you to press return, it will display the status of every interface as shown below.

```
Press RETURN to get started!  
*Mar 1 00:09:01.271: %LINK-5-CHANGED: Interface FastEthernet0/0, changed state to administratively  
down  
*Mar 1 00:09:01.583: %LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively  
down  
*Mar 1 00:09:02.271: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed  
state to down  
*Mar 1 00:09:02.583: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed  
state to down
```

Once you press enter, you will arrive at the **Router>** prompt. If the router has a startup config with authentication configured, such as in the case of most brand new ISRs, you will be prompted for a username and/or password before you will arrive at the prompt. For new ISRs cisco is the username and password. We will cover authentication later in the chapter. For now consider the prompt that you will see. The text before the greater-than sign (>) is the hostname of the device. By default **Router** or **Switch** is the default name depending on the device.

### IOS modes

The CLI of the IOS is divided into different modes or levels. Each mode serves a different purpose and has different sets of commands. It is important to be familiar with different modes that you will encounter in this book. Covering all the modes is out of the scope of CCNA.

The character after the hostname of the device tells you which mode you are in. When you first start a router and press enter, you are at the **Router>** prompt. The greater-than sign (>) tells you that you are in the **user exec mode** or **level 1**. This mode is mostly used to view statistics. You cannot view or edit configuration of the device from this mode. This mode also serves as the stepping-stone to the next mode, the **privileged exec mode** or **level 15**. At this level the prompt changes to the dollar sign (#). To go to the privileged exec mode from the user exec mode, type **enable** command on the prompt and press enter as shown below. Notice the change in prompt after the command is entered.

```
Router>enable
Router#
```

Congratulations! You just entered your first command on an IOS device.

To go back to the **user exec mode**, you can use the **disable** command as shown below:

```
Router#disable
Router>
```

To close the CLI session, use the **logout** command in any mode.

At the privileged exec mode you can view the configuration and statistics related to every component and process of the device but cannot make changes to the configuration. To be able to make changes to the configuration of the device, you will need to go to the **global configuration mode** using the **configure terminal** command in the privileged exec mode as shown below. Notice that the prompt changes to **Router(config)#** after you enter the command. **(config)#** tells you that you are in the global configuration mode.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```



## TUN MIN OO {BE-IT} Routing & Switching 200-120

In this mode, you can make changes to the configuration of the device. You must remember three things about the global configuration mode:

1. All changes affect the running config. These changes are not persistent after a reboot unless running config is saved to the startup config.
2. All changes have an immediate effect on the device.
3. The global configuration mode has sub-modes. While some changes can be made in the global configuration mode, changes to specific components, such as interfaces, must be done in dedicated sub-modes.

From the global configuration mode you can go to different sub modes to configure specific components. While most of the sub modes are beyond the scope of CCNA, a few of the modes that you will come across in the book are discussed in Table 3-4.

Sub-mode name	Purpose	Sub-mode prompt	Command to enter sub-mode
<b>Interface Configuration</b>	In this mode you can configure individual interfaces of the device. You can configure protocol, layer 3 addressing etc. in this mode.	Router(config-if)#	interface <interface-name>  Example:  Router(config)#interface fastEthernet 0/0  Router(config-if)#
<b>Line configuration</b>	In this mode you can configure the console, telnet and auxillary <i>lines</i> , which are used for exec sessions.	Router(config-line)#	line {con   vty   aux} <i>number</i>  Example:  Router(config)#line console 0  Router(config-line)#

<b>Routing Configuration</b>	In this mode you can configure the routing protocols.	Router(config-router)#	<code>router protocol [number]</code>  Example:  <code>Router(config)#router rip</code>  <code>Router(config-router)#</code>
------------------------------	---	------------------------	--

**Table 3-4** *IOS Sub-modes*

## IOS Editing and Help Features

While configuring a device running IOS, using the CLI is mostly about remembering the different commands and options. Cisco makes it easier to do this by providing various editing and help features. The help feature is a lifesaver. You can use a question mark (?) at any place to see a list of available commands or options, as shown below.

```
Router#configure ?
confirm      Confirm replacement of running-config with a new config
file
memory       Configure from NV memory
network      Configure from a TFTP network host
overwrite-network Overwrite NV memory from TFTP network host
replace      Replace the running-config with a new config file
terminal     Configure from the terminal
```

In the above output when a question mark (?) is entered after the **configure** command, a list of available options is displayed. Notice that **terminal** is one of the options. Another example is given below.

```

Router#?
Exec commands:
access-enable      Create a temporary Access-List entry
access-profile     Apply user-profile to interface
access-template    Create a temporary Access-List entry
alps               ALPS exec commands
archive            manage archive files
audio-prompt       load ivr prompt
auto              Exec level Automation
beep               Blocks Extensible Exchange Protocol commands
bfe                For manual emergency modes setting
call               Voice call
ccm-manager        Call Manager Application exec commands
cd                 Change current directory
clear              Reset functions
clock              Manage the system clock
cns                CNS agents
configure          Enter configuration mode
connect            Open a terminal connection
copy               Copy from one file to another
credential         load the credential info from file system
crypto             Encryption related commands.
ct-isdn            Run an ISDN component test command
-More-

```

In the above output, the numbers of options are more than the available screen size, hence the output pauses and you see the **-More-** text. At this point you can press space to see the rest of the output or press q to quit back to the prompt. A final example of the help feature is given below.

```

Router(config)#i?
identity  interface ip ipc
iphc-profile ipv6  ipx  irec-agent
isis      iua      ivr  ixl

```

In the above output notice that a question mark was entered after a single character. This causes IOS to display a list of options starting with that character. You can enter a question mark after multiple characters to see a list of options starting with those characters. For example, type **in?** at the above prompt will show a list consisting of **interface** option only. This brings up an interesting feature of the CLI. If you type a few characters which are unique to a command and press the tab key, the IOS will complete the command for you. In fact if you type the first few unique characters of the command, you need not

## TUN MIN OO {BE-IT} Routing & Switching 200-120

press tab or complete the command. IOS will understand which command you want. For example if you type **int** and press tab then IOS will complete the command. Another example is shown below.

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

Notice that the **configure terminal** command is executed at **conf t**. The IOS sees that the only command which starts with **conf** is **configure**, while **terminal** is the only option which starts with **t**.

Apart from these help features, the IOS provides some meaningful messages when you enter an incomplete or wrong command. Take a look at few of these messages shown below.

```
configure terminal
^
% Invalid input detected at '^' marker.
```

The above message tells that there is an error in the command marked by the caret sign (^). Because of the sign, it is easy to see that there is a typing mistake in the command.



```
Router(config)#interface
% Incomplete command.
```

The above message tells that you have entered an incomplete command. More options are needed with the command. In such a situation, you can use the question mark after the command to see available options.

```
Router(config)#s
% Ambiguous command
```

The above message shows that you have not typed enough unique characters. There are multiple commands that start with the characters that you have entered.

While using the CLI, these help features and messages are immensely useful, but you also need to know about a few key combinations that you can use while typing commands. Table 3-5 shows a list of these key combinations.

Table 3-5 IOS editing key combinations

Key or Combination	Purpose
<b>Left Arrow or Ctrl+b</b>	Move cursor one character back
<b>Right Arrow or Ctrl+f</b>	Move cursor one character forward
<b>Esc+b</b>	Move cursor one word back
<b>Esc+f</b>	Move cursor one word forward
<b>Ctrl+a</b>	Move cursor to the start of line
<b>Ctrl+e</b>	Move cursor to end of line
<b>Ctrl+d or Backspace</b>	Delete one character before the cursor
<b>Ctrl+w</b>	Delete one word before the cursor
<b>Ctrl+u</b>	Deletes the entire line
<b>Ctrl+z</b>	Leave configuration mode and go back to privileged exec mode
<b>Ctrl+p or Up arrow</b>	Shows the previous command entered
<b>Ctrl+n or Down arrow</b>	Shows the next command entered after up arrow/Ctrl+p has been used.

Another useful feature of the CLI is the **show history** command. This command lists the last 20 commands that you have entered in the session. An example is shown below:

```
Router#show history
enable
configure terminal
exit
show version
show run
show history
```

The number of commands that can be stored by the router in the history can be changed using the **terminal history size** command. You use the command to change the size of history from 0 to 256. An example is shown below:

```
Router#terminal history size ?
<0-256> Size of history buffer
Router#terminal history size 25
```

The configured size of the history can be confirmed by using the **show terminal** command as shown below:

```
myRouter#show terminal
Line 194, Location: "", Type: "XTERM-COLOR"
Length: 45 lines, Width: 202 columns
Baud rate (TX/RX) is 9600/9600
Status: PSI Enabled, Ready, Active, No Exit Banner, Automore On
Capabilities: none
Modem state: Ready
Special Chars: Escape Hold Stop Start Disconnect Activation
^x none - - none
Timeouts: Idle EXEC Idle Session Modem Answer Session Dispatch
00:10:00 never none not set
Idle Session Disconnect Warning
never
Login-sequence User Response
00:00:30
Autoselect Initial Wait
not set
Modem type is unknown.
Session limit is not set.
Time since activation: 00:00:31
Editing is enabled.
History is enabled, history size is 50.
DNS resolution in show commands is enabled
Full user help is disabled
Allowed input transports are pad telnet rlogin lapb-ta mop v120 ssh.
Allowed output transports are pad telnet rlogin lapb-ta mop v120 ssh.
Preferred transport is telnet.
No output characters are padded
No special data dispatching characters
```

In the above output, you can see that history is enabled and the history size is 50.

The last feature of CLI that you need to know about before proceeding is the **do** command. As you already know, commands can only be entered in particular modes. For example, **show terminal** command can be executed only in the user privilege mode and not in the configuration mode. If you are in the configuration mode or one of the sub-configuration modes, you will need to exit out of that mode and get back to the user privilege mode to execute that command. This can be inconvenient at times when you want to quickly verify something while configuring the device. To get around the need to exit back to the user privilege mode, you can use the **do** command followed by any privilege exec mode command to execute it from any mode you are. For example, the **do show terminal** command at the configuration mode will execute the **show terminal** command as if you were in the privilege exec mode. The example below shows what happens when the command is executed with and without **do**:

```
Router(config)#show terminal
```

```
^  
% Invalid input detected at '^' marker.
```

```
Router(config)#do show terminal
```

```
Line 194, Location: "", Type: "XTERM-COLOR"  
Length: 45 lines, Width: 202 columns  
Baud rate (TX/RX) is 9600/9600  
Status: PSI Enabled, Ready, Active, No Exit Banner, Automore On  
Capabilities: none  
Modem state: Ready  
Special Chars: Escape Hold Stop Start Disconnect Activation  
^x none - - none  
Timeouts: Idle EXEC Idle Session Modem Answer Session Dispatch  
00:10:00 never none not set  
Idle Session Disconnect Warning  
never  
Login-sequence User Response  
00:00:30  
Autoselect Initial Wait  
not set  
Modem type is unknown.  
Session limit is not set.  
Time since activation: 00:00:14  
Editing is enabled.  
History is enabled, history size is 20.  
DNS resolution in show commands is enabled  
Full user help is disabled  
Allowed input transports are pad telnet rlogin lapb-ta mop v120 ssh.  
Allowed output transports are pad telnet rlogin lapb-ta mop v120 ssh.  
Preferred transport is telnet.  
No output characters are padded  
No special data dispatching characters
```

In the above output, notice that the first command generated an error. But when it was executed with a preceding **do** command, it was executed.

### **3-3 Basic Configuration of Router and Switches**

In the previous section you became familiar with the IOS CLI and even used a few commands to move around it. In this section you will learn to configure a few basic administrative features. These features are not critical to the functions of the router in a network, but these features help make administering the router easier and secure. Remember that all of the commands in this section work on routers as well as switches.

- **Hostname** – You can set the name of the device with the **hostname** command in the global configuration mode. Setting the name of the router does not have any impact on the functions of the router. It will continue to perform normally irrespective of the name, but it is easier to manage and troubleshoot your network when you give the devices a meaningful name. The example



below shows how you can change the hostname. Notice the immediate change in prompt after the command is executed.

```
Router(config)#hostname Gateway
Gateway(config)#hostname myRouter
myRouter(config)#
```

- **Clock** – You can set the date and time on the router with the **clock** command in the privileged exec mode. Setting the correct date and time is a requirement for some advanced configuration and it helps when troubleshooting the device. The syntax of the command is **clock set hh:mm:ss day month**. An example is shown below:

```
myRouter#clock set 14:12:00 7 June 2011
```

- **Banners** – Banners are messages displayed to users who connect to the routers either through the various lines (console, vty and auxiliary) or through a dial-up. Banners are usually used to display a message prohibiting unauthorized access. There are many types of banners but you need to be aware of three kinds – the exec process creation banner, login banner and the message of the day (motd) banner. The motd banner is displayed first, followed by the login banner. These two banners are displayed before the login prompt. The exec process creation banner is displayed just after the login and before the prompt. You can use the **banner** command in the global configuration mode to create banners. The syntax for the command is **banner {exec | login | motd} delimiter text delimiter**. The *delimiter* is of importance here. It is any character which marks the start and end of the banner text. In the example below, the hash sign (#) is the delimiter.

```
myRouter(config)#banner motd #
Enter TEXT message. End with the character '#'.
Welcome to myRouter. No unauthorized access.
#
```

The above example is repeated below with the delimiter changed to the dollar sign(\$) and the entire command given in a single line.

```
myRouter(config)#banner motd $ Welcome to myRouter. No unauthorized access.$
```

The following output shows the banner displayed when an exec session is started.

```
myRouter con0 is now available
```

Press RETURN to get started.

```
Welcome to myRouter. No unauthorized access.  
myRouter>
```

### Securing access to the device

Routers and switches are the core of your network. A malicious user who gets access to these devices can cause network wide problems such as theft of data, loss of connectivity and more. Hence it is essential to secure access to your network devices. IOS provides two basic mechanisms for access restriction – **line passwords** and **enable password/secret**.

As you already know, an administrative (exec) session to an IOS device can be started using three methods – console, telnet/ssh and auxiliary. These are also referred to as **lines**. (The term **lines** is actually reminiscent of very old technologies so do not worry about why they are called such). The IOS allows you to configure a password on these lines so that anyone connecting to them is required to enter the password before being connected to the CLI. After reaching the CLI prompt, a user is at the user exec mode where they cannot even view the configuration. To be able to view or edit the configuration, the user needs to go to the privileged exec mode using the **enable** command. The IOS also allows you to configure a password that is required to go to the privileged exec mode. This is called the enable password or secret.

The enable password or secret can be configured using the following command in the global configuration mode:

```
enable {password | secret} password
```

There are four things that you should remember about the enable password and secret:

1. Enable secret is encrypted before being stored in the config while the enable password is stored as plain text. So anyone viewing the config will know the enable password.
2. If enable secret and enable password both are configured, the secret will be used always.
3. Enable secret and password cannot have the same value.
4. When using telnet or ssh to connect to the IOS, you cannot enter the enable mode if an enable secret or password has not been configured.

## TUN MIN OO {BE-IT} Routing & Switching 200-120

Here's an example of how they are configured:

```
myRouter(config)#enable password test
myRouter(config)#enable secret test123
```

After the enable password or secret is configured, notice the how the user is prompted for password when then enter the enable command in the user exec mode:

```
myRouter>en
Password: test123 (password will not be shown when typed on the device)
myRouter#
```

To configure a line password for console, you will first need to enter the line configuration mode for the console using the **line console** command in the global configuration mode as shown below:

```
myRouter(config)#line console ?
<0-0> First Line number
myRouter(config)#line console 0
myRouter(config-line)#
```

In the above output, I used a question mark at the end of the first line. The help output shows that 0 is the only option available. First thing to know here is that there can be multiple lines of a kind (example multiple telnet lines). Second you will need to specify the line number that you want to configure. In the case of console, there will always be only a single line, zero, available. So the command **line console 0** will bring you to the line configuration mode for the console line (notice the change in router prompt to (config-line)#).

In the line config mode, use the **password password** command to set a password for the line. After that you will need to use the **login** command to enable login with the password you just configured. The output below shows an example.

```
myRouter(config)#line console 0
myRouter(config-line)#password test
myRouter(config-line)#login
```

## TUN MIN OO {BE-IT} Routing & Switching 200-120

Now when someone tries to connect using the console, they will be prompted for a password as shown below.

```
myRouter con0 is now available
```

Press RETURN to get started.

User Access Verification

```
Password: test [password will not be displayed when typed]
myRouter>
```

Similar to how you configured the console password above, you can configure the password for the auxiliary line by going to the line mode for auxiliary using the **line aux 0** command. There is always a single auxiliary line in a device. The example below shows configuration required.

```
myRouter(config)#line aux 0
myRouter(config-line)#password test
myRouter(config-line)#login
```

Configuring the password for the telnet lines is no different, but you need to know two things before doing that:

1. Telnet lines are called vty lines because they are virtual unlike console and auxiliary
2. Each IOS device has a minimum of 5 vty lines (0 to 4). Some of them can have 15 or more.
3. You can configure all the vty lines together, in a group or one at a time. They need not have the same configuration.
4. A new telnet or SSH session will use the lowest available vty line. So there can be 5 telnet or SSH sessions to the device at any time.
5. Telnet or SSH sessions to the device will not be allowed unless a password has been configured and login is enabled.

To configure a password on line vty, you need to use the **password** and **login** commands in the line configuration mode. You can enter the vty line configuration mode using the **line vty *linenumber*** command. The following example shows the available number of vty lines:

```
myRouter(config)#line vty ?  
<0-4> First Line number  
myRouter(config)#line vty 0 ?  
<1-4> Last Line number  
<cr>  
myRouter(config)#line vty 0 4  
myRouter(config-line)#
```

The **line vty 0 4** command in the above example will enter the line configuration mode and you will be able to configure all the available vty lines at one time.

The example below shows a password configured for all the vty lines:

```
myRouter(config)#line vty 0 4  
myRouter(config-line)#password test  
myRouter(config-line)#login
```

Once the password has been configured and login enabled, the device will allow Telnet sessions to be initiated to the device. As you already know, Telnet is not a secure protocol because the session is transmitted in plain text and is vulnerable to snooping. To overcome this problem, SSH can be used. SSH encrypts the entire session but it requires encryption keys to start a session. By default IOS does not have these keys and hence a SSH session cannot be initiated. To generate those keys, you must first set the hostname and domain name of the device and then use the **crypto key** command as shown below:

```
myRouter(config)#hostname Gateway  
Gateway(config)#ip domain-name test.edu  
Gateway(config)#crypto key generate rsa general-keys modulus 1024  
% The key modulus size is 1024 bits  
% Generating 1024 bit RSA keys, keys will be non-exportable...  
Jun  9 00:43:43.599: %SSH-5-ENABLED: SSH 1.99 has been enabled  
Once the keys are generated, the vty line can be configured to accept SSH sessions using the following command:  
Gateway(config-line)#transport input ssh telnet
```

If you leave out the telnet option from the above command, only SSH will be allowed to the device.

One final thing you need to know about passwords is that the line passwords and the enable password is stored in the configuration as plain text. What this means is that anyone who comes across the configuration stored outside the device, can learn the passwords. To prevent this, the passwords can be encrypted using the **service password-encryption** command in the global configuration mode.

## **3-4 Configuring Router Interfaces**

While configuring interfaces of a switch are covered in Chapter 6, configuring the interfaces of a router is one of the basic things that you should know before forging further ahead. This is because unless the router is connected to the network, there isn't much it can do. Configuring the interfaces is easy and usually consists of only two steps. But before proceeding, you need to understand the interfaces and their numbering.

You will remember from earlier in the chapter that the number and type of interfaces are shown during the boot up. While there are many different types of interfaces that can be present in a router, the three types that you will encounter in the CCNA exam are Ethernet, FastEthernet and Serial. Some of these interfaces are built into the device while some are added as modules in available slots. The built-in

devices are said to be in slot zero; while module go into slot numbers starting from 1. Depending on the router the interfaces can be numbered simply as *type number* or *type slot/number* or in some high-end routers as *type router/slot/number*. Router, slot and number are numerical and start from 0. Some examples are:

Knowing the correct interface numbering is important because you need to know which interface to configure. Consider a situation where the Ethernet cable is plugged into the second interface of the second module while you configure the first interface of the first module! The question mark on the CLI can be of help in figuring out the numbering format when using the **interface** command in the global config mode. Take look at how the question mark helps in figuring out the format:

In the above output, the help output shows the different kinds of interfaces that can be configured.

In the above output, notice that only a single slot number, zero, is available

The above output shows that there are two FastEthernet Interfaces that can be configured, zero and one.

In the final output, the first built-in FastEthernet interface was selected. Once in the interface configuration mode (prompt changes to config-if), you can configure various parameters of the interface such as IP address, speed, duplex and in case of Serial Interfaces, protocols. For now, you will learn to configure an IP address and to enable the interface.

To configure an IP address, use the **ip address** command in the interface configuration mode. The command expects the IP address followed by a subnet mask as an argument as shown in the example below:

By default, all interfaces are in an **administratively down** status. What this means is, that all interfaces have the command **shutdown** applied to them by default and will not connect to the network unless they are brought up using the **no shutdown** command as shown below:



## TUN MIN OO {BE-IT} Routing & Switching 200-120

After this command is given, the router will bring up the interface and assume the IP Address you configured and will effectively be connected to the network on that interface. You can quickly verify connectivity at this stage using the **ping** and **tracert** command from the privileged exec mode as shown below:

While you already know about the ping command, the traceroute command might be new to you. The traceroute command uses the TTL field in the IP header to discover layer 3 devices between your device and a given host in the internetwork. Here the ping output shows that 192.168.1.10 is reachable from the router and traceroute command shows that it is the next hop device. Both of these outputs confirm that the router now has network connectivity and is able to function properly at all layers.

While FastEthernet interfaces usually required just the IP address and subnet mask, the serial interface might require some more configurations. However, that is discussed in detail in Chapter 11.

Certain poor network designs may require you to have a second IP address on an interface. While this is very inefficient, if you do run into a need to configure this, you will need to use the **secondary** keyword at the end of the **ip address** command. If you do not use the **secondary** keyword, the new one will replace the configured IP address on the interface. An example of adding a secondary IP address is shown below:

The secondary IP address can belong to the same subnet as the primary address or a different one.

Another optional command that you can use in interface configuration mode is the **description** command. This command will add a description text for the interface in the configuration. While this is not necessary for operation of the interface, it can be useful to have a short description containing the purpose of the interface. Some routers and most switches can have a lot of interfaces and it can be very difficult to decipher what the interface connects to. So it is recommended that you make a habit of adding description to all interfaces as shown below:

### **3-5 Gathering Information and Verifying Configuration**

While correctly configuring an IOS device is important, you should also be able to gather information and verify configuration easily. For this purpose, the IOS has many commands. You have already learned about the ping and traceroute commands in the previous section. This section introduces you to a range of show commands using which you can gather a lot of information and verify configuration and operation of the device.

The most prominent of the show commands is the show running-config command that displays the current running config of the device. Remember that running-config is different from startup-config. The example below shows the running-config of a device.



In the above output notice the various configurations from previous sections highlighted. In a single output, you can verify the entire configuration of the router.

Similar to the show running-config command, the show startup-config command shows the configuration stored in the NVRAM. If the running configuration is saved, the output of both commands will be the same as shown below:

Remember the system information displayed during the boot process? You can see that information again using the show version command. An example of the output of this command is shown below:

Notice that most of the information from the boot process is repeated in the above output. This command can be used to find the IOS version, number and types of interfaces, memory sizes etc.

The next show command that you should be familiar with is the show interfaces command. This command shows information related to an interface as shown below:

Some important lines in the above outputs are highlighted. Let us look at them and see what they mean. The most important line in the above output is the first line:

This line shows that the interface is up and the line protocol is up, which means that the interface is connected properly. Things are not always this good. You can have various some variations in the interface and line protocol status that indicate some problem. If the interface and line protocol both are shown as down, this indicates a problem with cabling or the interface and essentially a problem at the physical layer. If the interface is up but the line protocol is down, this indicates a problem at layer2 such as a framing or encapsulation problem. A third status that you can encounter is the administratively down status. This means that the shutdown command has been configured on the interface. You can use the no shut command in the interface configuration mode to bring it up.

The next three lines show the MAC address, IP Address, subnet mask and the MTU of the interface:

The next highlighted line shows the duplex and speed of the interface. As you can see, the interface is operating at 100Mb/s full duplex.

While troubleshooting, you might need to reset various counters such as count of packets input and output etc in the show interfaces output. This can be done using the clear counter command as shown below:

While the show interfaces command shows a lot of information, it is primarily geared towards layer 1 and layer 2 details. You can use the show ip interface command to look at IP related information as shown below:



While most of the above output is irrelevant to CCNA, you can see that the first four lines show the IP address, subnet mask, broadcast address and MTU of the interface. If you quickly want to see the IP address and status of all interfaces in the device, you can use the `show ip interface brief` command as shown below:

While `show interfaces` and `show ip interfaces` commands are useful to find a bunch of information, the `show ip interface brief` command is used to quickly find the status and address of every interface. Similarly, the `show protocols` command can be used to find this information as shown below:

The `show protocols` command is more useful if you have multiple layer 3 protocols running on the device.

When working with `show` commands, one useful feature that you can use is piping. Using pipes, you can search for specific lines in an entire output. Take a look at the example below:

In the above example, the output of `show running-config` is piped (using a pipe symbol!) and then the router is told to show lines from the output that include the word `address`. Similar to `include`, you can ask

the router to exclude certain words. In the example below, the word unassigned is excluded from the show ip interface brief command to show only interface that have an IP address:

The last pipe option that you need to know about, is the begin option. This option filters the output and shows the output starting from the line that contains the given word. Take a look at the example below:

In the example above, the output of show running-config is filtered to show only lines after the line containing the words "line con". Essentially you filtered the output to see the configuration of all the three lines.

Apart from the above discussed show commands, there are many that will be discussed throughout the book with relevant topics. For now, you should be comfortable using these commands and finding information.

## **3-6 Configuring DNS & DHCP**

### **Resolving Names on IOS**

When working with advanced configuration such as routing and access lists on IOS based devices, you will often refer to other devices. There are two ways to refer to another device – using the IP address or hostnames. Remembering and using IP address of various devices is difficult and often cumbersome to troubleshoot. Hence, IOS provides two ways to map names to IP address.

## TUN MIN OO {BE-IT} Routing & Switching 200-120

The first method is to use a DNS server that you might already have in the network. You can simply add the IP address of the DNS server using the **ip name-server** command as shown below:

Notice that I added two DNS servers in the above example. You can add as many as you want. The IOS will try to get a response from each server sequentially till it gets a response. Once you have added a DNS server, every time the device encounters a name, it will try to resolve it by querying the server.

The second method is to create a mapping of names to IP addresses on the IOS itself. These mappings constitute what is also known as the host tables.

Remember this does not make the IOS a DNS server, but simply creates a local list for the router. You can use the **ip host name ip\_address** command to do this as shown below:

In the above command, I mapped the name Router1 to the IP address 192.168.1.2. Now whenever the device comes across the name Router1, it will translate it to 192.168.1.2 as can be seen in the example below.

### Using DHCP

From the previous chapter, you would remember that DHCP is used to dynamically provide IP address to hosts in a network. An IOS device can be configured to receive as well as give out IP address. What this means is that the device can be a DHCP client as well as a DHCP server.

When you configure the device as a DHCP client, it takes an IP address from a DHCP server for its interface. To configure the interface to take the IP address from DHCP server use the **ip address dhcp** command in the interface configuration mode. After that, when the interface is brought up, it will start requesting for an IP addresses.

To configure the device as a DHCP server, you need to define a pool that consists of the subnet from which the device will give out addresses. Apart from the addresses, you can also configure other parameters such as DNS server and default gateway that can be sent to the client. To create the pool, use the **ip dhcp pool** *pool-name* command in the global configuration mode. This command will create the pool and bring you to the DHCP configuration mode (the prompt will change to dhcp-config). Here you can use the **network**, **dns-server** and **default-router** commands to define the subnet, DNS server and the gateway address as shown below:

One thing you may have noticed is that the entire subnet of 192.168.1.0/24 has assigned to the pool. However this also contains the addresses that has already been assigned to the router interface and the DNS server. You will need to exclude these addresses from the pool. This can be done using the **ip dhcp excluded-address** *ip\_address* command as shown below:

### **3-7 Saving, Erasing, Restoring and Backing up Configuration & IOS File**

#### **Saving, Erasing, Restoring and Backing up Configuration**

As you already know, any change to the configuration is made to the running-config, which is different from the startup-config that is loaded at the boot up. What this means is that if you do not save the running-config to the NVRAM as startup-config, all changes will be lost on reboot!

To save the running-configuration to NVRAM, you can use the **copy** command in the privileged exec mode. The copy command expects two parameters – the source and the destination. In this case, the

## TUN MIN OO {BE-IT} Routing & Switching 200-120

source is the running-config and the destination is the startup-config, so the command that you will need to use is **copy running-config startup-config** as shown below:

Now consider a situation where you made changes to the running-config but want to discard them. Remember that changes to the running-config are immediate. So either you can negate all changes one by one or simply copy the startup config to running-config using the **copy** command. This time, simply reverse the source and destination as shown below:

Apart from copying the config between running and startup, you can also copy the config to a TFTP server in the network for backup purpose. You need to backup the config so that you have a copy to use in case the router crashes and you need to replace it. You can again use the copy command by replacing the destination with tftp: as shown below:

Notice in the example above that after the command, you are prompted for the IP address of the TFTP server and the file name. You can press enter to use the default file name or provide a different one.

Similarly, you can copy the config from a TFTP server to the running-config by reversing the above command as shown below:

If you want to clear the configuration and start afresh, you can erase the startup-config and reload the router. To erase the startup-config, use the **erase startup-config** command in the privilege exec mode as shown below:

To reboot the device, use the **reload** command in the privilege exec mode as shown below:

If the configuration is not saved, the router will prompt you to save the configuration when you enter the reload command. Type no and press enter to not save the configuration and reload with an empty NVRAM. When the devices comes up, there will be no configuration on it and you can start over.

### Working with IOS files and IOS File System (IFS)

Similar to how the configuration can be copied around, the IOS file that is used by devices can be backed up or restored or simply changed using the copy command. Remember that IOS file is saved in the flash memory. To copy the file currently used by the router to a TFTP server, use the **copy flash:**

**tftp:** command as shown below:

To copy an image from the TFTP server to the flash memory, reverse the above command as shown below:

Notice the error in the above output? It shows that the file was not copied because there is no space left in the flash memory. So before copying the new file, you will need to free up some space. You can check the contents of the flash memory using the **dir** command in the privileged exec mode as shown below:

To free up some space you can delete some the existing files using the **delete** command. Since there is only a single file in the flash currently, it has to be deleted as shown below:

Now that you have free space, try the copy command again:



The two commands that you used above, **dir** and **delete** are examples of commands that are used to manage the IOS File System (IFS). Even **copy** is similarly a command to manage the IFS. There are various other commands that can be used to manage the IFS. Some of them are discussed below:

- **show file** – This command will give you few details such as file type and size about any file in the IFS. An example is shown below:
- **erase** – This command as you already know, can be used to delete the contents of the NVRAM.
- **format** – This command is used to erase ALL contents of the flash.
- **mkdir** – This command is used to create a sub-directory in the IFS. An example is shown below:
- **cd** – This command is used to move to a sub-directory or move back to parent directory on IFS. For example, **cd test** will take you inside the sub-directory created above.
- **pwd** – This command can be used to find the name of directory you are currently in. An example is shown below:

- [illegible]

In case you have enough free space in the flash memory, you can have multiple IOS files there. By default, the first file that the system encounters is used for booting. You can specify the IOS file that want the system to load to boot up, using the **boot system** command in the global configuration mode as shown below:

### **3-8 Password Recovery on a Cisco Router**

While working with IOS based devices, it is not uncommon to forget passwords and lock yourself out of the devices. While password recovery procedures differ from device to device, most Cisco routers have a similar process for recovering the password. Recovering passwords for Cisco switches is a little different and not covered in CCNA, so this sections looks at the password recovery procedure for routers only.

Before starting password recovery on a Cisco router, you need to understand two important things associated with the boot process:

- **The ROM monitor** – The ROM monitor, also called the bootstrap program, initializes the hardware, locates the IOS file and boots it. This mode can be used for troubleshooting and testing. If an IOS file is not found during the boot process, you will be dropped into the ROM monitor (ROMmon) mode. The prompt at this mode is `rommon #>`, where # is a number. At this mode there are very few commands available that essentially help in finding and fixing problems related to the boot up. It can also be used to copy an IOS file from TFTP to the flash.
- **Configuration Register** – This is a 16-bit value that is written to the NVRAM and controls aspects of the boot process. It can be set to change where bootstrap program looks for the IOS file, whether the startup config is loaded or not and even if the boot process should stop at ROMmon and not load the IOS file. While each one of the 16-bits has a different function there are two values that you need to remember – 2102 and 2142. 2102 is the default value, which means that the router will look for the IOS file in the flash memory and will load the startup config from the NVRAM. A value of 2142 means that the router will load the IOS file from the flash but will not load the startup config. The value of the config register can be seen in the output of **show version** command.

You probably have figured by now that the ROMmon and the configuration register play an important role in the password recovery procedure. The mains steps for the procedure are:

1. Boot into the ROMmon mode
2. Change the configuration register such that the startup config is not loaded
3. Boot into IOS
4. Go to the privileged exec mode and copy startup config to running config
5. Change the passwords
6. Save the running config to startup config
7. Change the configuration register back to 2102
8. Reboot

So essentially, you first get the router to load the IOS without the startup config, so that you can start an exec session and go to the privileged mode without a password. Once there, you load the startup config and change the password and then save back the config. One common mistake here is to not load the startup config. Remember that the running config is empty. If you simply configure the new password and save this config, it will override the startup config. Once the new passwords are saved in the startup config, you will need to change the configuration register back to normal.

## TUN MIN OO {BE-IT} Routing & Switching 200-120

Now let us look at the procedure in detail.

### Booting into the ROMmon mode and changing configuration register

To manually boot into the ROMmon mode, you will need to reboot the device and break the boot sequence. Pressing the Ctrl+Break key combination during boot usually does this. An example is shown below:



**Exam Alert:** In reality the break sequence differs from client to client and operating system to operating system. For example, when using OSX, you might have to use Cmd+b. As far as the CCNA exam goes, Ctrl+Break is the only option. I would suggest using Windows/Hyperterminal for practice.

Notice the second to the last line in the output where it shows that the boot was aborted due to user interrupt. That is where I press Ctrl+Break key combination and was brought to the rommon 1> prompt.

You can change the configuration register from this prompt using the **confreg** command as shown below:

Note that the configuration register value is preceded by 0x. This denotes that the value 2102 is a hexadecimal value. After changing the value, you can reset the device using the **reset** command here. When the device boots again, you do not have to interrupt the sequence.

### The rest of the procedure

After the device boots, the setup mode will begin. Type in *no* when prompted to exit the setup mode and enter the user exec mode. Once there, you can use the commands discussed previously in the chapter to get to the privileged mode, load startup config and change passwords. An example is shown below:

Now that the password has been changed in the startup config, you will be able to access the device once it boots back normally. But if you boot the router now, it will keep loading without the startup config because the configuration register is still set to 0x2142. To change the configuration register use the **config-register** command in the global configuration mode as shown below and then save the configuration again before rebooting:

### **3-9 Cisco Discovery Protocol (CDP)**

Cisco Discovery Protocol (CDP) is a proprietary protocol designed by Cisco to help in finding information about neighboring devices. Devices connected to each other exchange CDP packets to learn about each other. This can be useful in troubleshooting and documenting the network.

CDP is enabled on all interfaces of all Cisco routers and switches. You can disable CDP globally using the **no cdp run** command in the global configuration mode. It can be enabled again using the **cdp run** command. CDP can be disabled on an interface using the **no cdp enable** command in the interface configuration mode.

Each device running CDP sends out a packet every 60 seconds to its neighbors. The timers associated with CDP on a device can be seen using the **show cdp** command in the privilege exec mode as shown below:

In the above output you can see that CDP is sending packets every 60 seconds. Each neighbor will keep the information contained in a packet for 180 seconds. The timers can be changed using the **cdp timer** command and the **cdp holdtime** command in the global configuration mode as shown below:

As mentioned, earlier CDP can be used to troubleshoot as well as document a network. When you need information regarding devices directly connected to a device, you can check the neighbors learned by CDP using the **show cdp neighbor** command. An example is shown below:

The output shows that *myRouter* is directly connect to a device named *Switch3*. Each column in the output gives information regarding *Switch3*. Each column is explained below:

- **DeviceID** – This column gives the hostname of the directly connected device. In this case, the router is directly connected only to a single device named *Switch3*.
- **Local Interface** – This column shows the local interface of the device that is connected to the remote device. In this case, fa0/0 interface of *myRouter* is connected to *Switch3*
- **Holdtme** – This column shows the amount of time in seconds, that the local device will keep the information about the remote device, if no further packets are received from it. In this example, if *Switch3* does not send any more CDP packets before 172 seconds, it will be removed from the neighbor table of *myRouter*. The remote device advertises the holdtime.
- **Capability** – This column shows the capabilities of the remote device. The meaning of each letter in that column is shown at the beginning of the output. In this example, *Switch3* is shown as a Switch and has IGMP enabled on it.
- **Platform** – This column shows the device model of the remote device. In this output you can see that *Switch3* is a Cisco 2960 device.
- **Port ID** – This column shows the interface number of the remote device that connects to this device. In this example, *myRouter* is connected to fa0/8 interface of *Switch3*.

The **show cdp neighbor** commands provides brief information on all directly connect device. A more detailed information of a neighbor can be see using the **show cdp neighbors detail** command as shown below:



In the output above you will notice that apart from the information shown by the **show cdp neighbor** command, this output shows the IOS version, VTP and VLAN information as well as the duplex of the connection to the remote device. This output also shows the IP address of the remote device. This can be very useful if you want to connect to the remote device for troubleshooting. At this stage do not worry about VLAN and VTP. They are covered in the next chapter.

The exact same output can also be seen using the **show cdp entry \*** command. An example of the output is shown below again:

While the output of both **show cdp neighbors detail** and **show cdp entry \*** are the same, the latter gives you the option to just see the layer 3 protocol information or just the IOS version information from the remote device as shown below:

In the above output, notice that **show cdp entry \* protocol** gives only the IP address (layer 3 information) while **show cdp entry \* version** gives only the IOS version of directly connected devices.

CDP is a simple protocol that just works always. There will hardly be a need to troubleshoot CDP but in case you ever need to do that, you can use the **show cdp traffic** and **show cdp interface** commands. The **show cdp traffic** command displays information regarding the CDP packets sent and received. If

CDP traffic is not being sent or received or if there are errors, the output of this command will show that. An example of the output is shown below:

The **show cdp interface** command on the other hand will show CDP information related to each interface of the device. This command will show you if CDP is enabled on an interface or not and what are the timers associated with each interface. It will also show the status of the interface itself. An example of the output is given below:

One drawback of CDP is that it is a Cisco proprietary protocol and will not work if you have another vendors devices connected to a Cisco device. In such cases, you can use the **Link Layer Discovery Protocol** (LLDP). LLDP is an open standard protocol that does the same work as CDP but can be used between devices belonging to different vendors.

Not all Cisco devices currently support LLDP and it is not covered in CCNA, but you should know that it could be used in place of CDP.

## **3-10 Using Telnet on IOS**

Earlier in the chapter you learned that you can telnet into a Cisco device running IOS to manage it. It is also possible to telnet from a Cisco IOS device to another device. This can be useful when troubleshooting across multiple devices. In this section you will learn to initiate and manage telnet connections from IOS CLI.

To telnet to another device from IOS, use the **telnet** command as shown below:

Remember that if the remote device is an IOS device, it should have a line password and an enabled password/secret configured. Without these, you will not be able to telnet in or get into the privilege exec mode.

When you exit the telnet session using **logout** or **exit** command on the remote device, you will be back to the prompt of the device from where the session was initiated, as shown below:

You can telnet to multiple devices simultaneously. For that, you will first need to toggle back to the local device prompt from the remote prompt using **Ctrl+Shift+6** followed by the **X** key. For example, in the output below, I first telnet to the switch like before and enter its privilege exec mode. Then I press the **Ctrl+Shift+6** sequence following by **X**. The whole sequence will not be seen in the output but you will notice that I am back to the prompt of *myRouter*.

Though you are back to the first device, the telnet session to the remote device is still active, but in the background. Now you can initiate another telnet session to another device as shown below:

You can again leave this session active and go back to the prompt of the first device using the **Ctrl+Shift+6 X** sequence as shown below:

Now you have two telnet sessions active from the first device. You can see all active telnet sessions using the **show sessions** command:

## TUN MIN OO {BE-IT} Routing & Switching 200-120

The asterisk (\*) sign next to a session shows the most recent session. You can return to that session by pressing Enter twice. You can return to any session by typing the number of the session and pressing Enter. In the output below, I toggle between the two sessions. Notice how pressing enter twice or just entering the session number as a command takes me back to the telnet sessions. Also notice that I can toggle between the sessions and the first device using Ctrl+Shift+6 X sequence.

You can end a telnet session from the remote device using the **logout** or **exit** command. You can also close the telnet session from the originating device using the **disconnect<session\_id>** command.

In the output below, I close the first session using the logout command on the remote device whereas I close the second session using the disconnect command:

While managing a device, you can see who is connected to the device using the **show users** command as shown below:

```
myRouter#show users
```

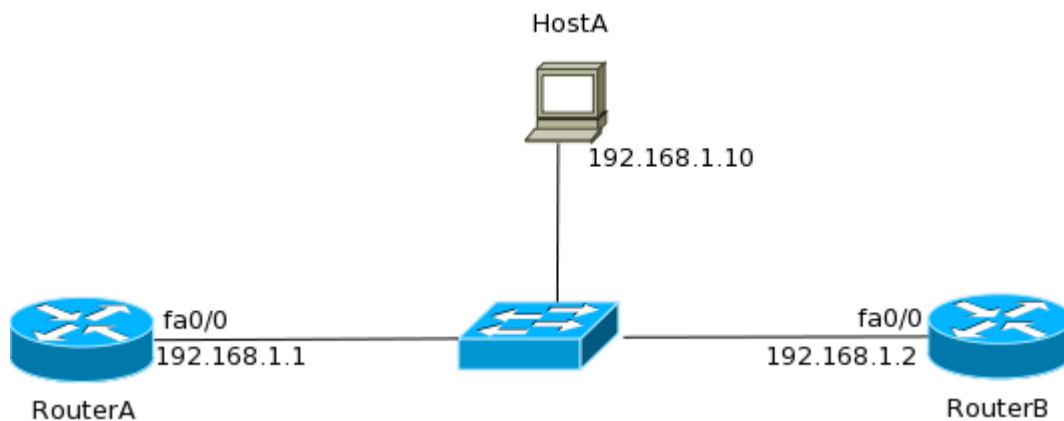
Line	User	Host(s)	Idle	Location
*194 vty 0	admin	idle	00:00:00	10.1.10.228
195 vty 1	admin	idle	00:00:00	10.1.10.18

The above output shows that two users are connected to the device using Telnet. The asterisks (\*) sign denotes the connection where the **show users** command was executed. The first connection is using line 194 and vty line 0 and the second is using line 195 and vty line 1. It is possible to disconnect someone's session from the device using the **clear line <line number>** command. In the output below, I have disconnected the second connection from the first session:

## 3-11 CCNA Lab #1

Your employer has installed two new Cisco routers in the network. Figure 3-5 shows the network layout. Your task is to configure the routers such that HostA can telnet into them to configure it further. Ensure correct hostnames and IP addresses are assigned. All passwords should be set to mypass123. When configuration is complete, save the configuration and back it up to the TFTP server running on HostA.

**Figure 3-5** Network Diagram for CCNA Lab #1



### Solution

1. Connect to the console port of RouterA and when prompted enter *no* to exit out of setup mode. Press Enter to go to the User exec mode.
2. Enter the privileged exec mode using the command **enable** and then configure the hostname and enable secret as shown below:
3. Configure the IP address on the interface as shown below:





763 bytes copied in 0.712 secs (1071 bytes/sec

## **Chapter 4 Introduction to IP Routing**

In the previous chapter, you configured a Cisco router such that it connects to the network and can be managed remotely amongst other things. Now it is finally the time to look at the single most important function of routers – IP routing. As you already know, routers look at the destination IP address of a packet and route it to the destination. Though the routing process itself is easy, building a list of networks, called the routing table, is not as easy. This chapter looks at the IP routing process itself and various ways to build to the routing table.

- 4-1 Understanding IP Routing
- 4-2 Static, Default and Dynamic Routing
- [4-3 Administrative Distance and Routing Metrics](#)
- 4-4 Classes of Routing Protocols
- 4-5 Routing Loops
- 4-6 Route Redistribution
- 4-7 Static and Default Route Lab

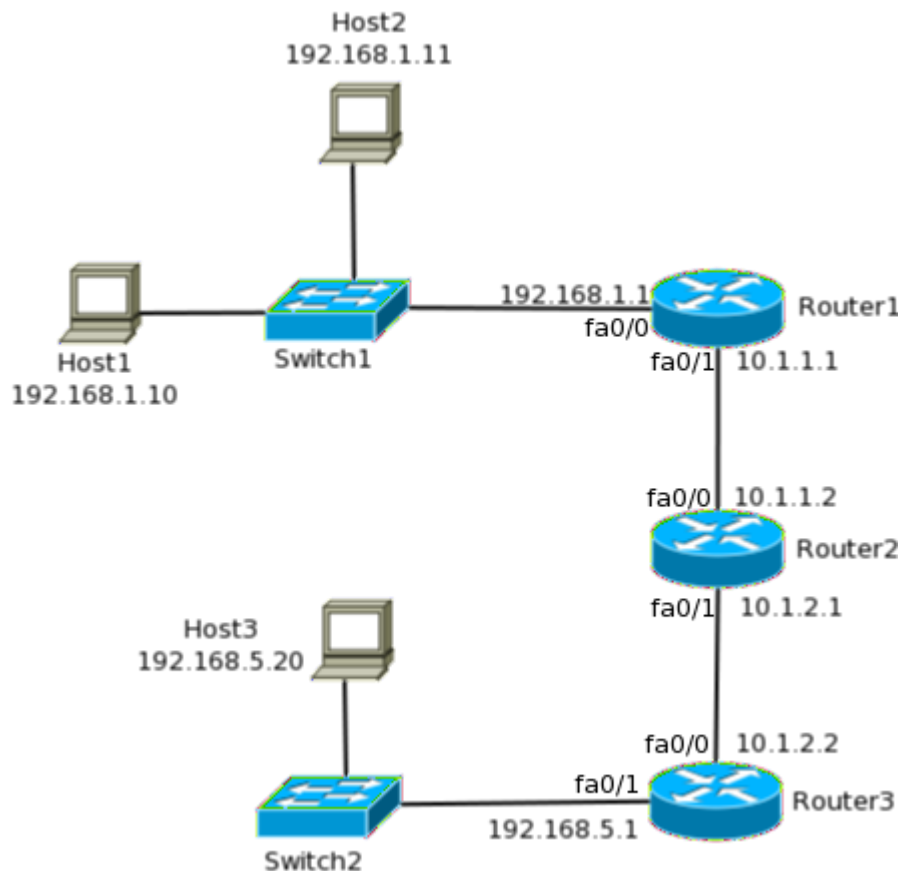
## **4-1 Understanding IP Routing**

In the simplest terms, IP Routing is the process of moving packets from its source to its destination across internetworks. To be able to route packets, a router must know at a minimum the following:

- Destination address
- Neighbor routers from which it can learn about remote networks
- Possible routes to all remote networks
- The best route to each remote network
- Be able to maintain and verify routing information

Unfortunately the process is not as simple as it sounds because it involves multiple protocols at multiple layers. To understand the complete process of how a packet moves from the source to the destination, consider the network shown in Figure 4-1.

**Figure 4-1** *Understanding IP Routing*



In the network shown above, when Host1 sends a TCP segment to Host3, the following happens:

1. The TCP segment is handed off to IP, which adds a header consisting of the source address, 192.168.1.10 and destination address 192.168.5.20 and hands off that packet to the next layer.
2. Using the subnet mask of the host, it is determined that the destination address lies in a remote network and hence the packet must be sent to the default gateway, 192.168.1.1. So Host1 sends out an ARP request to find the MAC address of Router1. When a response is received, it frames the packet with the source MAC address of Host1 and destination MAC address of Router1.
3. When Router1 receives the frame, it strips of the header and trailer and looks at the destination address in the IP header. Since the packet is not destined to Router1, it must be routed out.

4. It tries to match the destination address to a list of known networks, called the routing table. It finds that the destination network is reachable via Router2, so it frames the packet with the source MAC address of its exit interface (interface with the IP address of 10.1.1.1) and the destination address of Router2's interface.
5. When Router2 receives the frame, it repeats the strip and lookup process and frames the packet again before sending it to Router3. This time the MAC address of Router2's exit interface is the source address while the MAC address of Router3 is the destination address.
6. Finally Router3 looks at the destination MAC address and realizes that the destination network is directly connected. It finds the MAC address of the destination host and frames the packet using its own MAC address as the source while the MAC address of Host3 as the destination address. At last the frame is sent out and reaches the destination host.
7. At the destination, the frame is stripped and the destination IP address is verified. Then the IP header is stripped and the TCP segment reaches Layer 4 of the destination.
8. Now when Host3 needs to reply back to Host1, TCP will hand off the reply segment to IP.
9. IP will add a header consisting of a source address of 192.168.5.20 and a destination address of 192.168.1.10 and will send it to layer 2 for framing.
10. By the subnet mask of Host3, it is determined that the destination lies in a remote network. Hence the frame will need the MAC address of the default gateway as destination. If Host3 does not have the MAC address of Router3, it will send an ARP query to get it. Once Host3 has the MAC address, it will frame the segment and send it out to Router3.
11. Router3 strip the frame header and look at the destination IP address in the IP header. From its routing table, it will know that the packet needs to go to Router2. It will frame the packet with a source MAC address of its fa0/0 interface and the destination MAC address will be the address of Router2's fa0/1 interface and then send it out to the wire.
12. Router2 receives the frame and repeats process to send the packet to Router 1.

13. Router1 receives the frame from Router2 and removes the frame. By the destination IP address it knows that the packet belongs to a directly connected interface.
14. Since it received a frame from Host1 earlier, it has the MAC address of the host mapped to its IP address in the ARP table. The router uses that to create a frame with its fa0/0 interface's MAC address as source and Host1's MAC address as destination and sends the frame out the interface.
15. When Host1 receives the frame, it verifies the destination address, strips the frame and IP header and sends the TCP segment to layer 4.



**Exam Alert:** Remember that the source and destination IP address do not change throughout the process while the source and destination MAC address changes at each segment. You will see multiple questions about this on the CCNA exam! The MAC address is only locally significant and changes each hop.

The above steps show how a TCP segments moves from its source to its destination across an internetwork. The steps above assume that each router in the path knows where the destination network lies. But as you have seen in the previous chapter, a new router has no configuration and the router is not going to discover remote networks by itself. You will need to tell the router about the remote networks manually or configure it to learn the routes dynamically by talking to other routers.



**Note:** The network shown in Figure 4-1 will be used throughout the chapter. I strongly suggest you setup the above network and configure the basic connectivity. It will also allow you to practice everything learned in the previous chapter, once again.

## **4-2 Static, Default and Dynamic Routing**

### **Types of Routing**

To be able to route a packet, a router must know at least the following:

- Destination address to where the packet is destined. Layer 3 protocols such as IP take care of this.
- Neighboring routers from which remote networks can be learned of and packets can be moved on way to its destination.
- Routes to remote networks and a way to determine the best route to each of them.
- Way to learn, verify and manage routing information. Incomplete, incorrect or unstable routing information is worse than not having any routing information. If a router does not have routing information, it will drop the packets and let the source know. If a router has incorrect routing information, loops can form and bring down networks.

As you would have realized by now, the essence of routing is how the router learns about the remote networks. Routing information is stored in the routing table also called the **Routing Information Base (RIB)**. The RIB consists of routes to destination networks. Each route is a combination of the destination network address, subnet mask and the next hop towards the destination. There are three ways for a router to learn routes:

1. **Static Routing** – This is the method by which an administrator manually adds routes to the routing table of a router. This is a method for small networks but it is not scalable for larger networks.
2. **Default Routing** – This is the method where all routers are configured to send all packets towards a single router. This is a very useful method for small networks or for networks with a single entry and exit point. It is usually used in addition to Static and/or Dynamic routing.



3. **Dynamic Routing** – This is the method where protocols and algorithms are used to automatically propagate routing information. This is the most common method and most complex method of routing. Each routing protocol can have chapters or even whole BOOKS written about them. Most of them have one or more RFCs dedicated to them. In fact, the whole of the next chapter is dedicated to dynamic routing.

The following sections look at each of these routing types while implementing the first two types in our example network.

## Static Routing

When you manually add routes to the routing table, it is called static routing. There are advantages and disadvantages in using static routing. The advantages are:

1. There is no overhead in terms of CPU usage of the router as well as bandwidth between routers. When dynamic routing is used, packets are exchanged between routers and that uses bandwidth. That can be costly when they traverse across WAN links. The routers also need to process these packets and that consumes some CPU cycles as well.
2. It adds a certain degree of security since the administrator controls which routes the routers can know and learn.

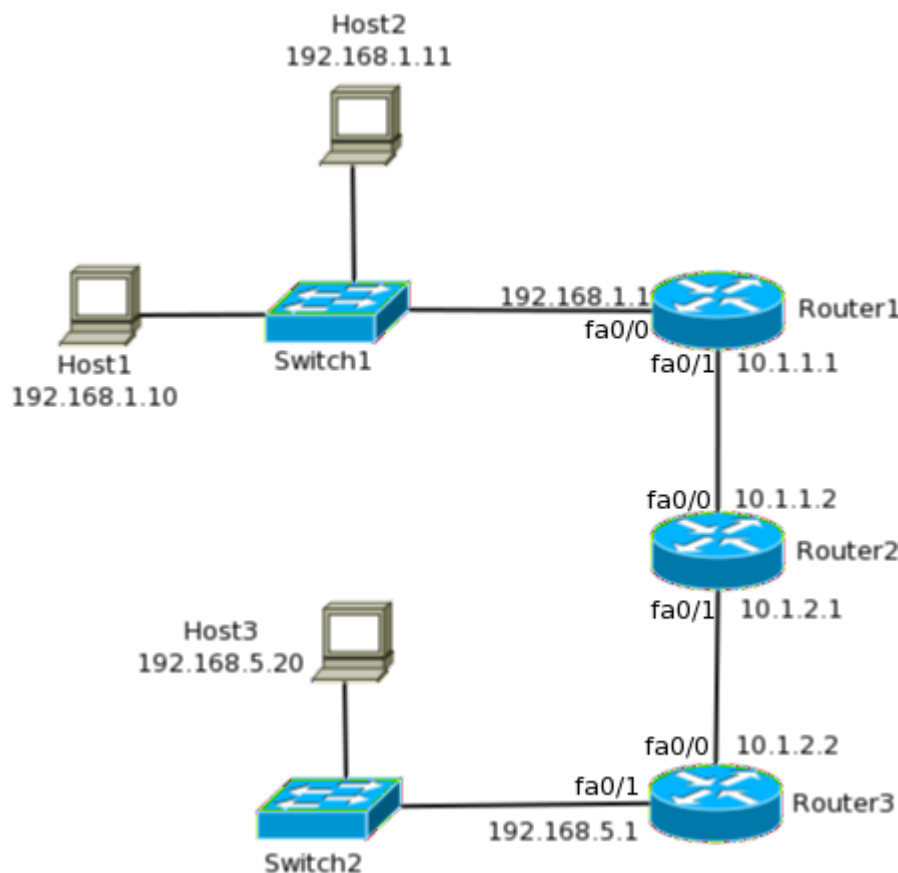
The disadvantages of static routing are:

1. The administrator needs to know the internetwork so well that he/she knows where each destination network lies and which is the next hop towards it.
2. Every change needs to be manually done on each router in the internetwork.
3. In large networks this can be unmanageable.

To add a static route, use the following command in the global configuration mode:

As you can see, the command is pretty simple. You need to specify the destination network address, its mask and the address of the next hop towards the destination. You can also specify the exit interface instead of the next hop address. Using the exit interface will cause the router to reply or ARP query and response from the next hop router and is not generally recommended.

**Figure 4-2** *Static Routing*



Let us configure our example network shown in Figure 4-2 (Figure 4-1 is repeated as Figure 4-2 so that you it is easier to understand), using static routing. To configure static routing, you need to look at the path traffic will taken from source to destination and back from destination to source. Each router in the path should know the source and destination network. So assuming our source is in network 192.168.1.0/24 (Host1) and our destination is in 192.168.5.0/24 network (Host3), let us look at the source to destination path, which is Router1->Router2->Router3.

1. Router1 does not know about the destination network. So we need to add a route. The next hop for Router1 towards the destination is Router2's fa0/0 interface. The route can be added using the following command:
2. Router2 also does not know about the destination network. So we need to add a route telling it that the next hop towards 192.168.5.0/24 is Router3's fa0/0 interface. The following command can be used to add the route:
3. Router3 knows the destination network since it is directly connected to it. So a route need not be added.

Following the path back from the destination to source, which is Router3->Router2->Router1:

1. Router3 does not know the 192.168.1.0/24 network, so a route should be added using the following command:
2. Router2 also does not know the 192.168.1.0/24 network, so a route should be added using the following command:

3. Router1, being directly connected to 192.168.1.0/24, knows about the network already.

To view the routing table and verifying static routing, you can use the **show ip route** command. The output from all three routers in our example is given below:

Though the output of the **show ip route** command will be discussed in detail later in the chapter and in the next chapter, here are a few things you need to know now:

1. The letter at the start of each line shows how the router was learned. The meaning of each letter is given at the beginning of the output as can be seen from the output from Router1. C stands for directly connected routes. These are the networks to which the router is directly connected. S stands for static routes. As you can see, the routes that you added are shown in lines that start with S.
2. You should verify the network and subnet mask in the output to see if you typed the correct information.
3. The IP address after “via” shows the next hop address for this destination.

The outputs show that all the routes that you added above have taken effect and traffic can flow between the 192.168.1.0/24 and 192.168.5.0/24 networks in both directions now. You may have noticed that Router1 still does not know about the network between Router2 and Router3 (10.1.2.0/24) and Router3 does not know about the network between Router1 and Router2 (10.1.1.0/24). Though it is not necessary for them to know about these networks, from a troubleshooting perspective it better to add routes for these networks also as shown below:

After these routes are added, the example network has complete reachability using static routing.

## Default Routing

Default routing can be considered a special type of static routing. The difference between a normal static route and a default route is that a default route is used to send packets destined to any unknown destination to a single next hop address. To understand how this works, consider Router1 from our example (Figure 4-2), without any static routes in it. When it receives a packet destined to 192.168.5.0/24 it will drop it since it does not know where the destination network is. If a default route is added in Router1 with next hop address of Router2, all packets destined to any unknown destination, such as 192.168.5.0/24 will be sent to Router2.

Default routes are useful when dealing with a network with a single exit point. It is also useful when a bulk of destination networks have to be routed to a single next-hop device. When adding a default route, you should ensure that the next-hop device can route the packet further, or else the next hop device will drop the packet.

Another point to remember is that when a more specific route to a destination exists in the routing table, the router will use that route and not the default route. The only time the router will use the default route is when a specific route does not exist.

The command to add a default route is same as that of adding a static route, but with the network address and mask set to 0.0.0.0 as shown below:

In our example network, the only exit point for the 192.168.1.0/24 and 192.168.5.0/24 networks is towards Router2. Hence, we can remove the static routes from Router1 and Router3 and add default routes as shown below:

Remember that since Router2 has multiple exists, you cannot use default routing there. It still needs the static routes.

Take a look at the routing table on Router1 and Router3 after the above changes:

In the above output notice that the static route to 0.0.0.0/0 is now seen in the routing table. Apart from that, the gateway of last resort is now the next-hop as specified in the default route.

A second way of adding a default route would be to specify the exit interface instead of the next-hop address. For example, on Router1, you can use the following command instead of the one used above:

**Router1(config)#ip route 0.0.0.0 0.0.0.0 fa0/0**

This tells the route to forward all packets, destined to unknown destinations, out fa0/0. While this will accomplish the same thing, the big difference is that a static route with an exit interface specified will take preference over a static route with next-hop specified. This is because the administrative distance of a route with exit interface is lower than the other one. Administrative distance is covered later in the chapter.

A third way of defining a default route is using the **ip default-network** command. Using this command you can tell the router to use the next-hop address of a known network as the gateway of last resort. For example, on Router1, you can use the following two commands to set the gateway of last resort:

The second command will cause the router to lookup the route to 10.1.2.0 and use 10.1.1.2 (next-hop address for 10.1.2.0) as the gateway of last resort.

The routing table will look as shown below, after the above two commands are entered:




The difference between using the **ip route** command and the **ip default-network** command for adding a default route is that the route added using **ip route** command is local and does not get propagated through a routing protocol, if one is enabled. The route added through the **ip default-network** command will get propagated by a routing protocol.

Another thing to remember is that prior to IOS version 12.4, the **ip classless** command was not enabled by default. You will remember from Chapter 2, that if the **ip classless** command is not used, the router will do classful routing and expect a default mask on each interface. A side effect of this command not being present is that if the destination network is not in the routing table, the router will drop the packet. If you are using default routing, it is possible that you do not have any specific routes in the table. So you must enable classless routing using the **ip classless** command for default routing to work.

## Dynamic Routing

Dynamic routing is when protocols, called routing protocols, are used to build the routing tables across the network. Using a routing protocol is easier than static routing and default routing, but it is more expensive in terms of CPU and bandwidth usage. Every routing protocol defines its own rules for communication between routers and selecting the best route.

Routing protocols are broadly classified as Interior Gateway Protocols (IGP) or Exterior Gateway Protocols (EGP). IGPs are used to exchange routing information within internetworks that fall under a single administrative domain (also called Autonomous Systems). EGPs on the other hand are used to exchange routing information between different autonomous systems. Common examples of IGPs are Routing Information Protocol (RIP), Enhanced Interior Gateway Routing Protocol (EIGRP) and Open Shortest Path First (OSPF). These are covered in detail in the next chapter. On the other hand, Border Gateway Protocol (BGP) is an example of EGP. It is the protocol used for routing information exchange on Internet. It is beyond the scope of CCNA, hence we will not cover it in this [BOOK](#) .

While the next chapter covers the IGPs in detail, the rest of this chapter is dedicated to basics of routing protocols that are necessary for you to understand before looking into specific protocols.

## **4-3 Administrative Distance and Routing Metrics**

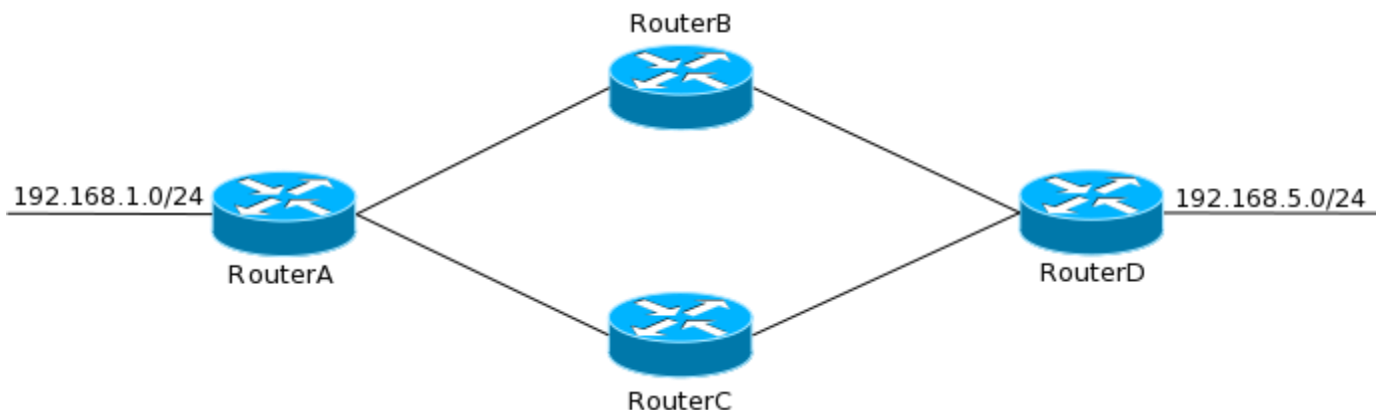
### **Understanding Routing Protocols**

While routing protocols are complex and have their own set of rules and commands, there are three basic things that you need to know before studying the individual protocols – **Administrative Distance**, **Classes of routing protocols** and **Routing loops**. The following sections look at these three topics in detail.

### **Administrative Distance**

To understand the importance of Administrative Distance (AD), take a look at the network shown in Figure 4-3.

**Figure 4-3** *Multiple paths to a destination*



Assume that you had one routing protocol between RouterA, RouterB and RouterD while another protocol between RouterA, RouterC and RouterD. (Yes, a router can run

multiple routing protocols at the same time). When both the routing protocols tell RouterA about the 192.168.5.0/24 network, which protocol's route should it choose?

The answer lies in the AD, which is the trustworthiness of routing information received by a router and it depends on the method or protocol by which that route was learned. What this means is, that each protocol has an AD value, which defines the trustworthiness of the routes it tells a router about. This value can be from 0 to 255, with a lower value being better. Each protocol has been assigned a default AD.

When Router1 receives information regarding the same network, 192.168.50./24, from two sources, it will compare the AD value of each source and the one with the lowest value will be selected.

On the other hand, if a single routing protocol was running on all routers, the routing protocol would see the multiple paths to the remote network and choose the best path depending on the **metric**. A metric, or cost, of a route is calculated differently by each protocol.

Table 4-1 shows the default AD value of various route sources.

**Table 4-1** *Default AD Values*

Route Source	Default AD Value
Connected Interfaces	0
Static Route	1
EIGRP	90
OSPF	110
External EIGRP	170
Unknown	255

As you can see from the table above, a connected route will be preferred over a static route, while a static route will be selected over any dynamic route. Similarly, an EIGRP route will be preferred over an OSPF route. Note that any route with an AD value of 255 will never be used.

It is important to remember the following when it comes to choosing routes:

1. When a routing protocol has more than one path to a destination, it will use metrics to present a route to the router.
2. When a router is presented with multiple routes to a destination, it will use AD to decide which one to use and will install that route in the routing table.
3. Finally when a router needs to route a packet, it will look at the routing table and use the route longest match prefix (subnet mask). For example, two routes are present in the routing table – 10.1.1.0/24 and 10.1.1.0/28, and a packet destined to 10.1.1.1 is received, the router will select the 10.1.1.0/28 route since it is the route with the longest match prefix.



**Exam Alert:** It is important to remember where AD is used and where metric is used. When it comes to actually routing the packet, the router will only look at the information in the routing table. AD and metrics are used to decide which route goes into the routing table only. You will surely see questions on the CCNA exam where they try to confuse you on how the route is selected.

You should note that the ADs given in table 4-1 are default ones and can be changed on any router easily. While changing the AD of dynamic routing protocols is out of CCNA context, you should know how to change the AD of static routes. You will recall that the command to add a static route is:

IOS provides two more options at the end of this command. The complete available command is:

The *metric* option can be any number between 1 and 255 and that number will become the administrative distance of the static route being added. For example, the following command will add a static route with an AD of 200:

You might ask why you would need to change the AD of a static route. Remember that AD is used to decide the best route when two different sources of routing information are presenting routes to the same destination. A static route is nothing but another source of routing information. By changing its AD to 200 in the above command, the static route is made less trusted than any route to 192.168.5.0/24 presented by dynamic protocols. This is commonly used to add a standby route in case dynamic routing fails. The static route will not be used unless no other source has a route to this destination.

The second option is the **permanent** keyword that will cause the route to be added to the routing table permanently irrespective of the fact that the next-hop is unknown. Without the **permanent** keyword, the route will be removed from the routing table (and not the configuration) if the next-hop is not in the routing table.

## **4-4 Classes of Routing Protocols**

Routing protocols are divided into the following three classes, depending on how they work:

1. **Distance Vector** – Distance Vector protocols are characterized by two things:
  - They use distance as a measure of the cost of a route. The number of hops in between a router and a destination network determines the distance.
  - They periodically send their entire routing table to the neighboring routers. The receiving router then merges its routing table with the received information based on AD and metrics. This process is called routing by rumor since the receiving router believes the information received from the neighbor.
  - Distance vector protocols are slower to **converge**. A network is considered converged when all routers in the network know about all destination networks. Distance Vector protocols are relatively easier to configure, manage and troubleshoot. However on the other hand, they consume a lot more bandwidth and CPU because they periodically send out the entire routing table, irrespective of the fact that nothing has changed in between the period. RIP is an example of a distance vector protocol.
2. **Link State** – Link state protocols are characterized by the following three things:
  - They form a neighbor relation with other routers before sharing the routing information. They do not send out routing information to the entire network

as in case of distance vector protocols. Information related to their neighbors are stored in a table.

- They only exchange connectivity related information or **link states**, unlike distance vector protocols that send out routing tables. This information is stored in a topology table to construct a full view of the network.
- Based on links states received, each router calculates the best path to every destination in the network. Each protocol has its own algorithm to calculate the best path.
- Link state updates are sent out only when there is a change instead of periodically as in case of distance vector protocols.
- Link state protocols converge faster than distance vector protocols. Link State Protocols are a little more complex to configure, manage and troubleshoot compared to distance vector protocols. OSPF is an example of a link state protocol.

3. **Hybrid** – Hybrid protocols use aspects of both distance vector and link state protocols. EIGRP is an example of hybrid protocol.



**Exam Alert:** Differences between the different classes as well as example of each class is an important topic in the CCNA exam.



## **4-5 Routing Loops**

A routing loop is a situation where a packet keeps getting routed between two or more routers because of problems in the routing table. In case of distance vector protocols, the fact that these protocols route by rumor and have a slow convergence time can cause routing loops.

To understand how routing loops can occur with distance vector protocols, consider the network shown in Figure 4-4.

**Figure 4-4** *Routing Loops*



When converged, all the routers in the network shown above will know about the 192.168.5.0/24 network. If RouterD loses connectivity to 192.168.5.0/24, it will remove the route to that network from its routing table. When RouterC receives the next periodic update from RouterD, it will know that the route to 192.168.5.0/24 is lost, and will remove it from its routing table. At this stage, RouterA and RouterB still think that 192.168.5.0/24 is reachable via RouterC.

While RouterC waits to send out the periodic update, if RouterB sends its own update, it will contain 192.168.5.0/24 as a destination network. Since RouterC does not have that network in its routing table, it will assume that it is a new destination and RouterB knows about and will install the route to that network, pointing towards RouterB. After this, the

periodic update from RouterC will contain the 192.168.5.0/24 network and RouterB will assume that it knows of all the networks contained in that update!

Now when RouterB receives a packet destined to 192.168.5.0/24, it will forward it out to RouterC. When RouterC receives that packet, it will see that 192.168.5.0/24 is towards RouterB and will send it back. This loop will continue till the IP TTL value in the packet header reaches zero and one of the routers drops it.

To prevent against such routing loops, distance vector protocols have some checks in place. These checks are discussed in the following sections.

### Maximum Hop Count

Without checks in place, the wrong routing information can spread throughout the network. To prevent this, protocols such as RIP have a maximum hop count. For RIP this value is set to 15. Any route with more than the maximum hop count is deemed unreachable and will not be used. In the above scenario, the original hop count of 192.168.5.0/24 on RouterB was 2. After RouterA lost the connectivity and RouterC learned the wrong information, it would see 192.168.5.0/24 at 3 hop counts. When RouterB gets this update back from RouterC, it will add 1 to the hop count and make it 4. This cycle will go on. Without a maximum hop count in place, this will go on. This phenomenon is called **counting to infinity**. Without maximum hop count in place, the increasing hop count will cause the routes to be deemed unreachable, and will be removed from the routing table causing the loop to be resolved.

### Split Horizon

The split horizon rule states that routing information learned from one interface cannot be advertised back to that interface. With this rule in place in the above scenario, RouterB would have never advertised 192.168.5.0/24 network back to RouterC since that's where the route originated. Hence a routing loop would never occur. By default split horizon is enabled for RIP and EIGRP.

## Route Poisoning

Route poisoning uses the maximum hop counts to stop network loops. When a router loses a route, it advertises that route with a hop count of more than the maximum hop count. The receiving router now finds the destination network unreachable and advertises it ahead as such. It also sends the update back towards the source router to ensure that the route is now poisoned in the entire network. This process is called **poison reverse**.

In the above network when RouterD loses 192.168.5.0/24, it would advertise the route to RouterC with a hop count of more than the maximum hop count. RouterC in turn will update RouterB. This is the route poisoning process. RouterC also sends the poisoned route back to RouterD to ensure that the whole network is in sync. This is the poison reverse process.

## Hold Downs

Routing protocols implement timers to allow lost routes to recover or to switch to the next best route to the same destination. These timers are called hold down timers. This is typically useful in case of links going down and coming back up rapidly (this is called **flapping**). One such route going in and out of the routing table can cause loops and stop the network from converging. Hold down timers also prevent changes which affect a route that was recently lost.

In the above example, a hold down timer would have prevented the update from RouterB from affecting RouterC immediately after the route to 192.168.5.0/24 was lost. In the meantime, RouterC would have updated RouterB about the lost route.



**Exam Alert:** All the loop prevention methods are important topics in the CCNA exam

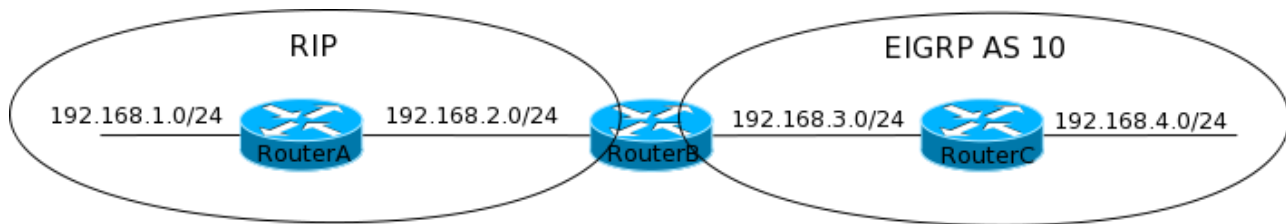
## **4-6 Route Redistribution**

While you are not really going to redistribute routes as a part of CCNA, it is important to know what it is. Simply put, route redistribution is the process of distributing routes learned from one source to another. This is useful when networks are expanding, or are merging, or in a phase of transition.

For example, assume that RIP is being used in a growing network. Beyond a hop count of 15, it will become impossible to use RIP. In this situation, you will need to switch to another routing protocol. While switching, two protocols would need to co-exist in the network while maintaining complete reachability. Redistribution of routes from RIP to the new protocol and vice versa can achieve this.

Another example where you would need to use redistribution is when a company acquires another and their networks need to merge. If both the networks were using different routing protocols, redistribution between these protocols can provide full connectivity with the least amount of effort.

**Figure 4-5** *Route Redistribution*



A Few important points that you should remember about route redistribution are:

1. The routing protocol receiving the redistributed routes will mark them as external. External routes are less preferred than Internal routes
2. Routes can only be redistributed at routers that run both the routing protocols. For example, Figure 4-3 shows RouterB running EIGRP and OSPF both. In the network shown, routes can be redistributed on RouterB only.
3. It is possible to redistribute between two different processes or AS of the same protocol. For example, if you have two EIGRP AS running on a Router, you can redistribute between them.
4. Static and Connected routes can also be redistributed
5. Only routes present in routing tables can be redistributed. For example, if a static route points to an unknown next-hop, it will not be present in the routing table and cannot be redistributed.
6. When redistributing routes, you have to ensure metric compatibility. For example, EIGRP metrics can be large numbers while any metric above 15 is considered invalid in RIP. In such cases, you have to tell the receiving routing protocol how to translate the metrics.

## **4-7 Static and Default Route Lab**

### **Lab 4-1 Static and Default Route**

Problem: In the network shown in Figure 4-4, configure each router such using static and default routes such that there is complete connectivity through the network.

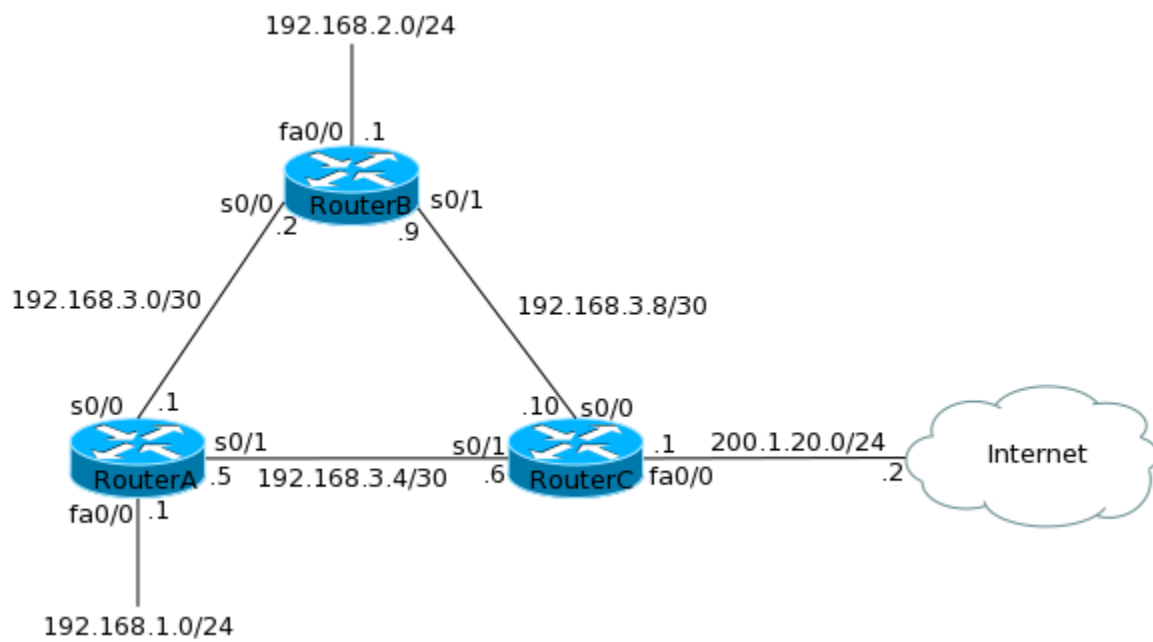
The initial configuration of each router is given below. Lab note: The DCE side of your DCE/DTE back to back cable plugs into the interface with the clockrate configured. If you neglect this, the lab will not work as the interface will not stay up.

#### **RouterA**

**RouterB**

**RouterC**

Figure 4-6 Lab 4-1





## Solution:

To provide full connectivity across the network, each router will require static routes to the different networks attached to the routers. To reach the Internet, all routers will require a default route. The solution is shown below:

## Verification:

To verify, first check the routing table of each router:

You can also use ping command to verify connectivity across the network as shown below:



## Summary

While this chapter was light reading compared to the previous chapters, it lays the foundation for the next chapter where you will learn the traits of individual protocols and how to configure them. It is essential that you are able to use static and default routing well before heading into routing protocols.

It is also important to understand the difference between administrative distance and metrics and where each is used.

## **Chapter 5 Routing Protocols**

The previous chapter introduced you to the IP routing. It also discussed the basics of dynamic routing. This chapter continues from where the last one ended and looks at three routing protocols – RIP, EIGRP and OSPF in detail. In this chapter you will learn about the traits of each of these protocols, how to configure and how to troubleshoot them.

- 5-1 RIPv1 & RIPv2
- 5-2 Configuring RIPv1 & RIPv2
- 5-3 Verifying and Troubleshooting RIP
- 5-4 Enhanced Interior Gateway Routing Protocol (EIGRP)
- 5-5 Configuring EIGRP
- 5-6 Verifying and Troubleshooting EIGRP

- 5-7 Open Shortest Path First (OSPF)
- 5-8 Configuring OSPF
- 5-9 Verifying and Troubleshooting OSPF
- 5-10 EIGRP and OSPF Summary & Redistribution Routes
- 5-11 Lab 5-1: RIP
- 5-12 Lab 5-2: EIGRP
- 5-13 Lab 5-3: OSPF

## **5-1 RIPv1 & RIPv2**

### **Routing Information Protocol (RIP)**

Although the new version of the CCNA exam 200-120 does not cover RIP, we want to touch on it for its historical value. This way you understand some of the basic characteristics of it and how a hybrid protocol such as EIGRP took some distance vector based features from a true distance vector protocol. So simply read through this to have a basic foundation of RIP and do not worry about it from a test perspective.

As you already know, RIP is a distance-vector protocol. In fact, it is the only distance vector protocol that is widely used today. There are two versions of RIP that can be

used – RIP version 1 (RIPv1) and RIP version 2 (RIPv2). To make it easier to understand, this section first looks at RIPv1.

RIPv1 was originally defined in RFC 1058 and is a classful protocol. Hence, it does not advertise subnet mask information and assumes the default subnet mask based on the class of the network.

When a router starts up, it recognizes the connected networks and adds them to its routing table as connected routes (denoted by C in the routing table). When RIP is enabled, it will broadcast the routing table using UDP port 520. All neighboring routers that have RIP enabled will get this broadcast update and add the routes received in the update to their routing table. Each of these neighbors will in turn broadcast out their routing tables. This will cause the routing tables across the network to converge.

Being a distance-vector protocol, RIP has the following characteristics:

1. It sends out its entire routing table every 30 seconds.
2. It uses hop counts as metric and has a maximum hop count limit of 15.
3. It implements split horizon, route poisoning and holddown timers to prevent routing loops.
4. It has high convergence time

### RIP Timers

Notice that there are two timers mentioned above. RIP actually uses 4 different timers. To understand these timers consider the network shown in figure 5-1. If RIP is enabled on all the routers, after convergence, all the routers will know the 192.168.5.0/24 network

**Figure 5-1** *Understanding RIP Timers*



Now take a look at the four timers used by RIP:

- **Route update timer** – RIP sends broadcasts out the entire routing table. This interval sets the interval between these updates.
- **Route invalid timer** – If a router does not hear any updates about a particular route for certain duration, it will consider that route as invalid. The invalid timer determines this duration. When a route becomes invalid, the router will send out poisoned routes to its neighbors. By default this value is 180 seconds. In the network above, if RouterC loses connectivity to RouterD, it will not hear about the 192.168.5.0/24 network. It will wait 180 seconds before considering the route as invalid and sending out poisoned routes.
- **Holddown timer** – When a route becomes invalid, it enters into a holddown state. In this state the route will remain in the routing table and packets will be forwarded towards the destination but the router will not accept any updates regarding this route unless the update contains a metric equal to or better than the existing metric. The holddown timer determines the duration of the holddown state. By default this duration is 180 seconds. This state is useful to ensure that flapping routes do not cause instability. In the network above, when RouterB gets the poisoned route from RouterC, it will put the route to 192.168.5.0/24 in the holddown state for 180 seconds. If RouterC regains connectivity to RouterD and updates RouterB, the route will be removed from the holddown state.
- **Router flush timer** – Once a route becomes invalid, it is put in a holddown state. While in the holddown state, the route is still in the routing table and will remain so for the duration specific by the flush timer. Once this timer expires, the route is flushed out of the routing table. By default this timer is 240 seconds and starts at the same time as the invalid timer. Hence the flush timer must be more than the invalid timer. In the above example, RouterA, RouterB and RouterC will remove the route to 192.168.5.0/24 60 seconds after it was marked invalid.

The timers can be a little confusing. To make it easier to understand, remember that:

1. Invalid timer and Flush timer both start when the router receives an update for a route. Each time an update is received, the timers are reset back.



2. If an update for a route is not heard for the duration of the invalid timer, it is marked invalid and the holddown timer is started.
3. While the route is in the holddown state, the router will not accept an inferior route for that destination. Inferior route is an update with a metric worse than or equal to the existing one.
4. The route will be removed when the flush timer expires.

In the above network, route to 192.168.5.0/24 becomes invalid 180 seconds after RouterC loses connectivity to RouterD. At this stage, 60 seconds are left in the flush timer. Hence 60 seconds after the route became invalid; it will be removed from the routing table. As you can see, it takes a total of 240 seconds or four minutes for a lost route to be removed from the routing tables across the network.

## **5-2 Configuring RIPv1 & RIPv2**

### **Configuring RIPv1**

Configuring RIP is pretty easy and consists of the following two steps:

1. Enable RIP globally using the **router rip** global configuration command. This command will bring you to the routing configuration mode as shown below:

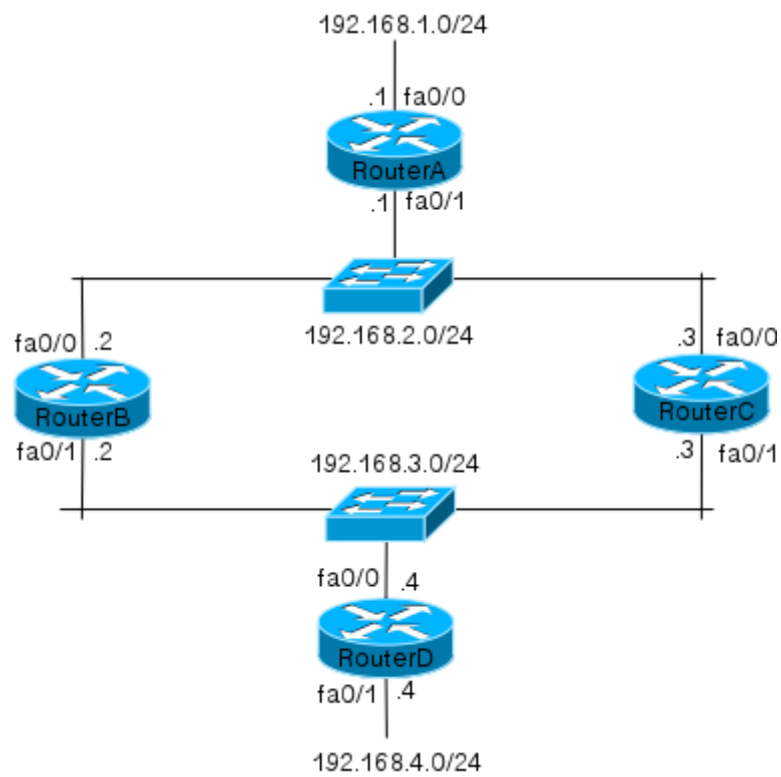
1. Tell the router which networks to advertise using the **network** *<network>* command in the routing configuration mode as shown below:

Remember that the network command is used to tell the router that **connected** routes you want to advertise. Any routes learned from other routers will automatically be advertised out. Since RIPv1 is being used, the **network** command will accept classful networks only. As soon as the **network** command is given, RIP will begin sending out updates as well as processing updates received from neighbors.

The network shown in Figure 5-2 will be used for the rest of the RIP sections.

Now that you know how RIP works and how to configure it, let us configure the network shown in Figure 5-2 to see effect of RIP on the routing table. For this example, we will enable RIP on RouterA, RouterB and RouterD only. RouterC will be configured in one of the sections ahead. The configuration required on the three routers is shown below:

**Figure 5-2** *RIP example*



Now take a look at the routing table on each of the three routers to see the effect:

In the above output, note the lines that start with R. The R signifies that these routes were learned from RIP. In output from RouterA, notice that the route to 192.168.4.0/24 network was learned from RIP. The 120/2 in the line shows that the administrative distance of the route is 120 (default RIP AD) and that the destination network is two hops away. The next hop towards 192.168.4.0/24 is 192.168.2.2, which is RouterB. Similarly you will notice that each router now knows about every subnet in the network. You may have noticed that compared to static or default routing, configuring RIP was easier and faster. Now when there is a change in the network, the routing table will automatically get updated across the network.

### **RIP version 2 (RIPv2)**

RIPv1 was one of the earliest routing protocols and was very popular back when it was created. With evolution in networking standards, RIP was found lacking in many places. Hence RIPv2 was developed in 1993 and standardized under RFC 2453. While RIPv2 is also a distance-vector routing protocols and fundamentally similar to RIPv1, there are some difference in the way it works. Table 5-1 shows the differences between RIPv1 and RIPv2.

**Table 5-1** *Differences between RIPv1 and RIPv2*

<b>RIPv1</b>	<b>RIPv2</b>
<b>It is a classful protocol and does not send subnet masks in routing updates</b>	Is a classless protocol and sends subnet masks in routing updates
<b>Uses broadcast to communicate with neighbors</b>	Uses multicast to communicate with peers. Multicast address 224.0.0.9 is used.
<b>RIPv1 does not support authentication</b>	RIPv2 supports authentication
<b>Does not support VLSM</b>	Supports VLSM

Remember that apart from the differences given in Table 5-1, RIPv2 is similar to RIPv1 with a maximum hop count of 15 and same timers as RIPv1. It also implements the same loop prevention techniques as RIPv1. The configuration for RIPv2 is same as RIPv1 but requires the addition of **version 2** command in the routing configuration mode. RouterA, RouterB and RouterD from our previous example can be configured to use RIPv2 as shown below:

Take a look at the routing tables of these routers after the change:

You will notice that the routing table output is same irrespective of the RIP version used. The output will only differ between the two protocols if the default mask is not used for the given class. In such a case, you will notice that when RIPv2 is used, the subnet mask is correctly seen on the neighbor while in case of RIPv1 the neighbor assumes the default subnet mask. To show this difference in the routing table, I temporarily added a 192.168.20.0/25 network on RouterA and advertised it using RIPv1. The output of the routing table from RouterB is shown below:

In the above output notice that the route to 192.168.20.0 network has a mask of /24 instead of /25. When the version was changed to 2, notice the routing table output on RouterB:

Notice that the mask for the 192.168.20.0 is correct displayed at /25 when RIPv2 was used.

### Stopping RIP updates on an Interface

As soon as RIP is enabled, it will start sending and receiving updates on interfaces. Many situations require you to stop RIP from sending updates out an interface. An example of such a situation is when an interface connects to the Internet. You do not want your routing updates to go out to the Internet. In such situations, you can use the **passive-interface** *interface* command in the routing configuration mode to stop RIP from sending updates out that interface. This command stop RIP from sending updates but it will continue to receive updates on that interface.

In our example network, we do not need to send RIP updates out interface fa0/0 on RouterA and interface fa0/1 on RouterD. We can stop updates going of of these interfaces using the following commands:

Remember that we did not configure RouterC earlier? Let us configure RouterC to run RIP across both its networks as shown below:



After the above configuration, the routing table on RouterC looks as shown below:

The output above is similar to what was seen in RouterB. So why did we not configure RouterC earlier? Take a look at the routing table of RouterA after we enabled RIP on RouterC:

In the above output notice that RouterA's routing table has two paths listed to 192.168.4.0/24 and 192.168.3.0/24. Similarly, RouterD has two paths listed for 192.168.1.0/24 and 192.168.2.0/24 as shown below:

To explain this behavior, consider what happened when RouterC started advertising its routes to RouterA and RouterD. Till that point, RouterA has only one way to reach 192.168.3.0/24 and 192.168.4.0/24 with hop counts of 1 and 2 respectively. When RouterC advertised its routes to RouterA, it also advertised the networks 192.168.3.0/24 and 192.168.4.0/24 with hop counts of 1 and 2. At this stage, RouterA has two paths to the same destination and both paths have the same metric. As you already know, when a routing protocol has multiple paths to a destination it compares the metric to decide which path to use. In this same we have two **equal cost paths**. When a routing protocol has two or more equal cost paths, it will use both the paths and the traffic will be **load balanced** across both the paths. Hence in the above outputs you see two paths for the destination networks.

RIP can load balance between 4 equal cost paths by default. The older codes of Cisco IOS support load balancing across a maximum of 6 equal cost paths while the newer codes support load balancing across a maximum of 16 equal cost paths. You can change the default value of 4 using the **maximum-paths** *number* under the routing configuration mode.

### **5-3 Verifying and Troubleshooting RIP**

Know how to verify and troubleshoot a protocol or feature is as important as knowing how to configure it because configurations do have errors and assuming that everything

is working correctly can lead to major network problems. The following three commands are used to verify and troubleshoot RIP:

- `show ip route`
- `show ip protocols`
- `debug ip rip`

The **show ip route** command has been covered in the previous chapter and earlier in this chapter. Eventually a complete and correct routing table across the network is the best verification of a routing table. The other two commands are covered below.

### Using show ip protocols command to verify and troubleshoot RIP

As you already know, the **show ip protocols** command helps verify routing protocols running on the router. An example of the output of this command from RouterB of our network is shown below:

Notice in the above output that RIP is being used on the router and it is routing for the 192.168.2.0 and 192.168.3.0 networks. RIPv1 is being used on both the interfaces and updates are being sent every 30 seconds. It also shows the fa0/0 is a passive interface.

While the output is very useful in verifying the configuration, it is also useful for troubleshooting. Looking at the above output, it is fairly easy to figure out what has been configured. For example, the output shows that it is routing for networks 192.168.1.0 and 192.168.2.0 so the following network commands should be present under the routing configuration:

In addition to that, the fa0/0 interface is shown as passive, hence the **passive-interface fa0/0** command is also present in the configuration. A quick look at the **show ip interface brief** command will show you the interfaces of the router and their IP address:

Comparing the above outputs, it is easy to see that RIP is running on the correct interfaces and networks.

Another important thing, which the output shows you, is that it is sending and receiving RIPv1 updates. You can confirm the versions across the routers in the network to rule out version mismatch if routing updates are not seen on few routers.

### Using debug ip rip command to troubleshoot RIP

The **debug ip rip** command displays routing updates on the console as soon as they are sent or received. This output is useful to see if the updates are being sent to and being received from the neighbors or not. The following example shows the output of the command on RouterA:

The first output shows the routes received from 192.168.2.2 (RouterB) and the metric associated with each route. It also shows that version 1 is being used. The send output shows RouterA sending out the update for 192.168.1.0. Remember it will not include the 192.168.3.0 and 192.168.4.0 networks in the updates because of split horizon. It is also not sending out updates on fa0/0 because we have configured it as passive.

The last output shows the updates received from 192.168.2.3 (RouterC) and metrics associated with the routes.

The outputs of **debug ip rip** command are also useful to see the loop prevention techniques in action. To trigger the loop prevention techniques, interface fa0/0 was shutdown on RouterD.

You will remember that a router will wait for the duration of the invalid timer before it sends out poisoned routes and it will wait for the duration of the flush timer before removing the routes. By default the invalid timer is 180 seconds and the route is flushed 60 seconds after invalid timer. The following output shows the poisoned routes received from RouterB and RouterC on RouterA:

RouterA#

Sixty seconds after the poisoned routers, the update from RouterB no longer contains the 192.168.4.0 route. This shows that RouterB has flushed that route from its table:

Once the interface is brought back up on RouterD, the 192.168.4.0 network is seen again in the update received from RouterB on RouterA:

As a final example, I changed the version on RouterB to 2. This will cause a mismatch with RouterA and RouterD sine they are configured for version 1. The following output on RouterA is seen when it receives the update from RouterB:

On the other hand, the following output is seen on RouterB when it receives updates from RouterA:

The two outputs above clearly show that there is a version mismatch and you will need to use the **version** command in the routing configuration mode to fix it.

## **5-4 Enhanced Interior Gateway Routing Protocol (EIGRP)**



While RIP is a good protocol to use in a small and simple network, its disadvantages become obvious in large and complex networks. Few problems associated with RIP in such networks are:

1. It has a maximum hop count of 15. This means that RIP cannot be used on a network spanning more than 15 routers
2. It uses hop count as the sole metric even where multiple paths are available. Hop count is not a suitable metric since links can have varied bandwidths. For example, in Figure 5-2 if the link between RouterA and RouterB has a bandwidth of 1Mbps while the link between RouterA and RouterC has a bandwidth of 128Kbps, RIP will still consider both links equal since the hop count is same. It is usually desirable to use the better link before the slower one.
3. It has a high convergence time.

Due to these disadvantages, other routing protocols such as EIGRP should be considered in place of RIP.

EIGRP is a Cisco proprietary classless routing protocol that is essentially an enhanced distance vector protocol or a hybrid protocol. It takes various features of distance vector protocols and link state protocols to overcome the disadvantages associated with distance vector protocols while retaining the simplicity associated with them.

EIGRP inherits the following features of a distance vector protocol:

1. It has a maximum hop count limit of 100 by default and can be increased up to 255.
2. It uses routing-by-rumor mechanism.
3. Implements loop avoidance techniques such as split horizon.

It inherits the following features of a link state protocol:

1. It discovers neighbors and periodically checks their status
2. Instead of periodic updates, it send updates when change occurs

EIGRP has some features that make it stand out from other protocols such as RIP and OSPF. While discussing each of them is out of scope of CCNA, the most important ones are listed below:

1. Supports multiple routed protocols such as IPv4, IPv6, Appletalk, IPX etc via protocol-dependent modules (PDMs)
2. Is a classless protocol and supports VLSM/CIDR.
3. Supports summaries and discontinuous networks
4. Uses neighbor discovery.
5. Utilizes Reliable Transport Protocol (RTP) for communication between neighbors
6. Uses Diffusing Update Algorithm (DUAL) for best path selection. This algorithm considers multiple metrics for the purpose.

The following sections look at the various features of EIGRP in detail.

## Multiple Network Protocol Support

EIGRP provides support for multiple Network layer protocols such as IPv4, IPv6, IPX and Appletalk. It supports these protocols through the use of **Protocol Dependent Modules (PDMs)**. Separate tables are maintained for each network layer protocol for which EIGRP is being run. While you will be learning about EIGRP in respect to IPv4 only, it is important to remember that EIGRP supports multiple protocols. The only other routing protocol that supports multiple network layer protocol is **Intermediate System-to-Intermediate System (IS-IS)**.

## Neighbor Discovery and Communication

One of the most important features that EIGRP adopts from link state protocols is neighbor discovery and adjacency formation. Unlike distance vector protocols, link state protocols and EIGRP will not exchange routes with just anyone. Routers running EIGRP will first discover other routers running EIGRP by sending out **Hello packets**. These packets are multicast to address 224.0.0.10. When two routers receive Hello packets from each other, they compare the following information found in the packet:

1. **Autonomous System (AS) Number** – A router can belong to one or more EIGRP autonomous systems. As you know, an AS is group of devices under a single administrative domain. EIGRP adjacency can be formed only between routers that belong to the same AS. The hello packets contain the AS number to which the sending router belongs to.
2. **Identical Metrics (K values)** – EIGRP uses various metrics to calculate the best path. These metrics are also called K-values. A router can be configured to use some or all of these metrics. Two routers cannot form an adjacency if they have been configured to use different sets of K-values. These metrics are discussed in detailed later in the chapter.

EIGRP routers form adjacency because routing updates are not sent out periodically via EIGRP like normal distance vector protocols. So EIGRP needs a way to know when a new neighbor has joined or when a previously known neighbor went down. The only time EIGRP sends out the entire table is when a new neighbor is discovered.

Another benefit of adjacency is that it helps divide the routers into different autonomous systems. Routers belonging to different autonomous systems will not form adjacency and hence will not share the routing tables. This is very beneficial in a large network where routing tables can become huge. Dividing the routers into different autonomous systems can help reduce the routing table size.

EIGRP uses a proprietary protocol called **Reliable Transport Protocol (RTP)** to manage communications between neighbors. This protocol is designed to provide very reliable communication between neighbors. RTP uses both multicast and unicast to deliver updates quickly and tracks acknowledgement of updates.

EIGRP will send out updates whenever there is a change in the network. This update is sent to multicast address 224.0.0.10. Each update is assigned a sequence number and neighbors have to acknowledge receipt of each update. Using sequence numbers an EIGRP router is able to track which neighbors have acknowledged an update. If an acknowledgement is not received from a neighbor, EIGRP will send the same update to this neighbor using unicast. If an acknowledgement is not received after 16 unicast

messages, the neighbor is declared dead. This process is often referred to as **reliable multicast**.

When EIGRP sends out an update, loss of packets can cause routing tables in the network to get corrupted. Thus the reliability offered by RTP is very important to EIGRP.

### **Diffusing Update Algorithm (DUAL) and EIGRP metrics**

EIGRP uses **Diffusing Update Algorithm (DUAL)** for selecting the best path to remote networks. The main features of DUAL are:

1. Support of VLSMs
2. Recovering lost routes dynamically.
3. Determining the backup route and using it when the main route is lost.
4. Finding alternate routes if a route is lost and no backup route is found.
5. Using various metrics to determine the best routes.

DUAL is responsible for the fast convergence time in EIGRP. In fact, the convergence time of EIGRP is possibly the fastest amongst all routing protocols. This fast convergence is achieved because all EIGRP routers maintain a copy of the network topology. If the best route goes down, a router simply scans the topology table and selects a backup route. If a backup route is not found in the topology table, the router will reach out to its neighbors to find an alternate path.

Another feature that differentiates DUAL is the use of multiple metrics to calculate the best path instead of using single metric like most other routing protocols. EIGRP can use the following four metrics to calculate the best path:

1. Bandwidth (also called path bandwidth value)
2. Delay (also called cumulative line delay)
3. Load
4. Reliability

By default it uses only bandwidth and delay to calculate the best path, but it can be configured to also use the other two metrics. Remember that an adjacency will not form between two routers that have been configured to use different metrics.

A fifth element, **maximum transmission unit (MTU)** size, is also required in some situations such as redistribution but is never used in EIGRP calculations. This value represents the smallest MTU value between the router and the remote destination network.

To find the best path to a network, DUAL uses the different metrics of each path in an algorithm to compute the cost of the path. The path with the lowest cost is considered the best. The exact formula used to calculate the path using the metrics is out of scope of the CCNA exam.

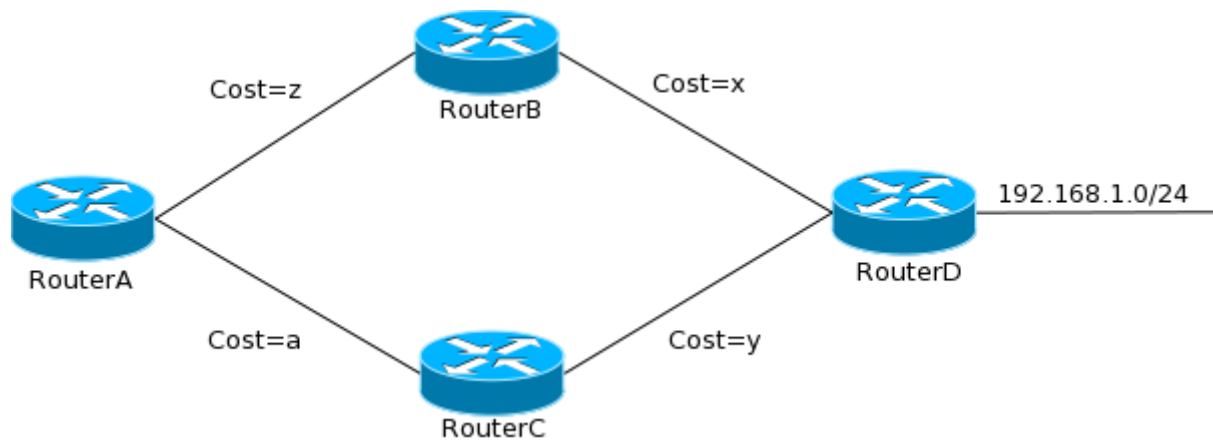
### Route Discovery and Best Path Selection

So far, you have learned about RTP and DUAL and how routers form adjacencies. EIGRP is one protocol that believes in finding and storing as much information about the network as possible. As a router learns about neighbors and forms an adjacency, it stores the details of each neighbor in a table called the **Neighborship** or **Neighbor table**.

After adjacencies have formed, routing tables are exchanged between neighbors. These tables contain information regarding remote networks and path to them. This information is stored in a table called the **Topology table**. The information received from the neighbor consists of the following:

1. Remote network's address
2. Remote network's subnet mask
3. Next hop to the remote network
4. Cost to the remote network

**Figure 5-3** Reported and Feasible distance



While the first three items are self explanatory, the cost is something that needs further explanation. The cost reported by the neighbor is the cost from the neighbor to the destination network. It does not include the cost from the receiving router to the neighbor (advertising router). This cost is known as the **Reported** or **Advertised distance**.

When the receiving router adds the cost between itself and the neighbor to the reported distance, the resulting cost is known as the **feasible distance**.

To further understand this, consider the network shown in Figure 5-3. Assuming that EIGRP is running on all routers in the network, RouterB learns about the 192.168.1.0/24 network from RouterD. The cost (feasible distance) from RouterB to the destination network is **x**. When RouterB advertises this network to RouterA, it will report the cost as **x**. Here **x** is the reported distance for RouterA. The cost between RouterA and RouterB is **z**. RouterA will add this cost to the reported distance to find the feasible distance or the total cost to the destination network. For the network 192.168.1.0/24, the feasible distance on RouterA is **x+z**.

It is important to understand here that the receiving router, RouterA in our case, has to add the cost between itself and the advertising router (RouterB is our case) to get the total cost to the destination network. This total cost is known as the feasible distance.

For each destination network that a router learns about, it will select the path with the lowest cost. This path is then sent to the router to be added to the routing table and is known as the **Successor**.

To select a backup path, the router compares the feasible distance of the successor with the reported distance of other available paths to the same destination network. If the reported distance of the other path is less than the feasible distance of the successor, the other path is marked as a backup path and is known as the **feasible successor**.

To further understand how a feasible successor is selected, assume that the successor route to 192.168.1.0/24 network from RouterB in Figure 5-3, is the RouterA->RouterB->RouterD path. The feasible distance of this path is the sum of z and x ( $z+x$ ). On the other hand, RouterA learns of another path to the 192.168.1.0/24 network from RouterC. The reported distance of this path is y. This route will be considered as a backup route or the feasible successor only if y is less than the sum of z and x ( $y < z+x$ ).

EIGRP will store up to six feasible successors to a single destination in the topology table.

This section introduced a lot of new and important EIGRP concepts and it is important to remember them. The list below summarizes important topics discussed in this section above:

1. **Neighbor Table** – Stores information about routers with whom an adjacency has been formed.
2. **Topology Table** – Stores information about every route and destination network learned from neighbors.
3. **Reported Distance** – The cost from the advertising router to the destination network.
4. **Feasible Distance** – The cost from the receiving router to the destination network. This is the reported distance plus the cost of path between the receiving and the advertising router.
5. **Successor** – The best route to a destination network.

6. **Feasible Successor** – The backup routes to a destination network. The reported distance of a route has to be less than the feasible distance of the successor for it to be marked as a feasible successor.



## **5-5 Configuring EIGRP**

EIGRP configuration is divided into two modes – the router configuration mode and the Interface configuration mode. The global configuration such as AS number and networks to advertise are configured in the router configuration mode, the Interface specific configuration such as metrics and timers are configured in the Interface configuration mode.

The steps to Enable EIGRP and define the networks to be advertised are similar to that of RIP and can be done in the following two steps:

1. Enable EIGRP globally using the **router eigrp** *as-number* global configuration command. This command will bring you to the routing configuration mode as shown below:

The AS Number can be anything from 1 to 65535 but has to be same on all routers that need to form adjacency.

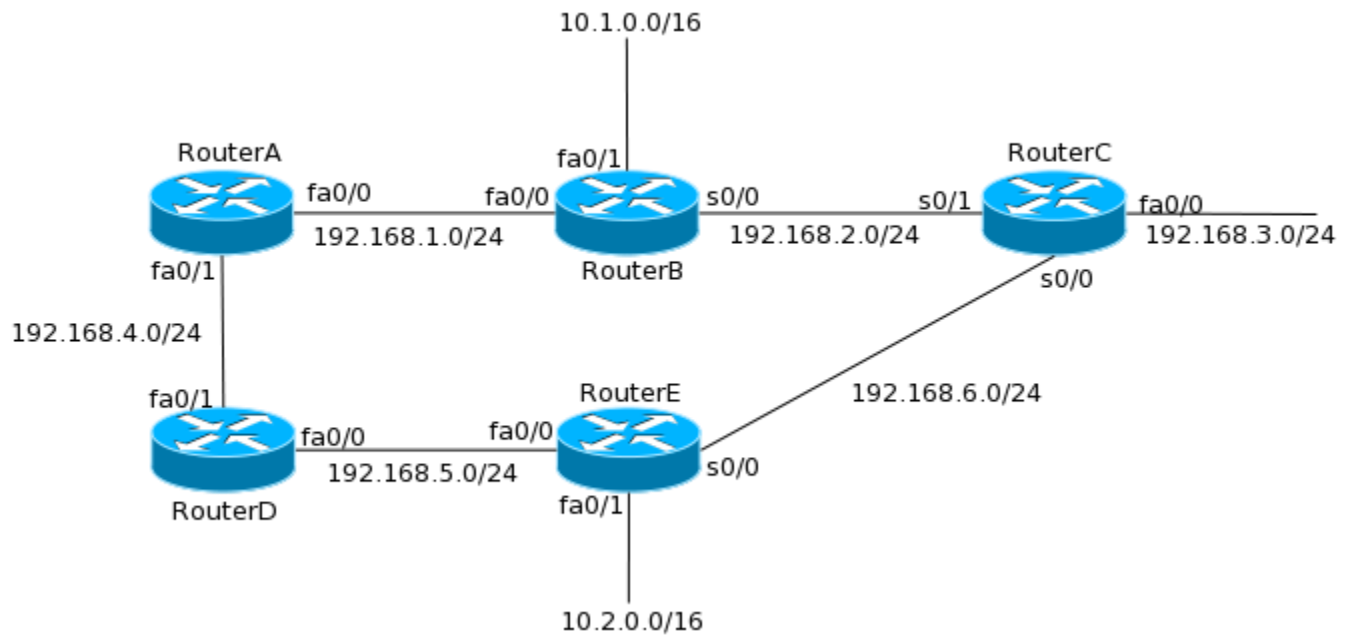
1. Tell the router which networks to advertise using the **network** *<network>* command in the routing configuration mode as shown below:

EIGRP gives the option to use wildcard masks when configuring EIGRP but for CCNA we will use classful networks only when defining networks in EIGRP.

Remember that the network command is used to tell the router the **connected** routes you want to advertise. Any routes learned from other routers will automatically be advertised out. As soon as the **network** command is given, EIGRP will begin sending out hello packets to the network and wait for hello packets from the neighbors.

Figure 5-4 will be used for the rest of EIGRP sections in this chapter.

**Figure 5-4** *EIGRP example network*



Now that you know how EIGRP functions and how to configure it, let us configure the network shown in Figure 5-4 to see EIGRP in effect. We will not configure RouterE in this section.

The configuration required on the four routers to get EIGRP working is shown below. All routers will be configured to be EIGRP AS 10.

Now look at the routing table of each router to see the effect. Remember that lines starting with D denote a route learned from EIGRP.



## Stopping EIGRP updates on an Interface

As soon as EIGRP is enabled on an interface, it will start sending and receiving hello packets on its interfaces. Many situations require you to stop EIGRP from sending hello packets out an interface or forming an adjacency via that interface. An example of such a situation is when an interface connects to the Internet. You do not want your routing updates to go out to the Internet. In such situations, you can use the **passive-interface** *interface* command in the routing configuration mode to stop EIGRP from sending hello packets out that interface.

In our example network, we do not need to send EIGRP updates out interface fa0/1 on RouterB. We can stop updates going out off these interfaces using the following commands:

Notice that the behavior of the passive-interface command is different in EIGRP than in RIP. In RIP, updates will not be sent out a passive interface but will continue to be received. EIGRP on the other hand, will not send or receive updates on a passive interface.

## Multiple Autonomous Systems

As you already know, AS is used to group routers into a single administrative domain in EIGRP. Routers belonging to different AS will not form an adjacency and thus will not exchange routes.

Having a single AS across a large network can cause it to have a complicated topology and routing table. In such networks, convergence can slow down during network changes. To mitigate this, large networks should be broken into multiple ASes.

Routing information is not shared between different ASes by default. Dividing a large network into multiple ASes will cause incomplete routing tables. To mitigate that, routes are **redistributed** between the ASes at points where they intersect. While redistribution is out of scope of CCNA, you should remember the following when it comes to EIGRP redistribution:

1. Normal EIGRP routes are called **internal** routes and have an administrative distance (AD) of 90. On the other hand, redistributed routes are called **external** routes and have an administrative distance of 170. Even routes redistributed between two EIGRP ASes are treated as external routes
2. When redistributing from one EIGRP AS to another, the metrics are not changed. This is because EIGRP understands its own metrics! On the other hand when redistributing between different routing protocols, you need to tell the receiving routing protocol how to treat the metrics. This is because EIGRP will not understand metrics from OSPF and similarly OSPF will not understand metrics from EIGRP.

### VLSM Support and Summarization

EIGRP propagates subnet masks along with routes in its updates. This enables it to support Variable length subnet masks (VLSM). As you already know VLSM helps conserve subnets through use of subnet masks. This also helps EIGRP support discontinuous subnets. A discontinuous network is one that has two subnets of a classful network connected together by another classful network.

An example of discontinuous network can be seen in Figure 5-4 where 10.1.0.0/16 and 10.2.0.0/16 networks are separated by 192.168.x.0/24 networks. Remember that RIPv1 does not support such networks.

EIGRP by default does not support discontinuous networks, but can be configured to do so. To understand the problems that arises when a protocol does not support discontinuous networks, let us configure RouterE to use EIGRP:

Let us verify the routing table on RouterE first:

Now to see the problem associated with a routing protocol not supporting discontinuous networks, take a look at the routing table of RouterA:

Notice that there is only one route to the 10.0.0.0/8 network pointing towards RouterB whereas in our network we have two 10.x.0.0/16 networks.

Similarly on RouterD, there is only a single 10.0.0.0/8 route pointing towards RouterE:

So traffic destined to 10.2.0.0/16 network will be routed to RouterE from RouterD but to RouterB from RouterA.

This happens in EIGRP because by default EIGRP automatically summarizes the networks at classful boundaries. Which means that RouterB and RouterE by default



advertise the 10.x.0.0/16 network as 10.0.0.0/8 network. This behavior of EIGRP can be changed to support discontinuous networks by using the **no auto-summary** command in the global configuration mode. The followings commands disable auto summary on all routers in our network:

The above changes will cause EIGRP to reset all adjacencies and form them again, creating a small window when the routing tables will not be updated. After the adjacencies come back up, the routing table on RouterA will look like the following:

In the above output notice that there are now routing entries for both 10.1.0.0/16 and 10.2.0.0/16 networks, both pointing towards the correct next hop. Similarly, the routing table on RouterD has entries for each of those networks:

As you saw in this section, discontinuous networks can cause routing problems but EIGRP can support them with a little change.

### **EIGRP load balancing and maximum hops**

Like RIP, EIGRP can load balance across a default of 4 equal cost paths. It can be configured to load balance across a maximum of 6 paths (for older IOS versions) and 16 paths (for IOS versions 12.2(33) and above). The difference between EIGRP and RIP load balancing is that EIGRP can be configured to load balance across unequal cost paths also.

EIGRP load balancing can be seen in our setup on RouterC. You may have noticed that RouterC has two paths to the 192.168.4.0/24 network and both paths use similar links. Hence RouterC will load balance traffic destined to 192.168.4.0/24 network as can be seen in its routing table:

To see the effect of metrics on EIGRP load balancing, I reduced the bandwidth on the s0/0 interface to 100 Kbit/sec. This caused the cost of the route to 192.168.4.0/24 via 192.168.2.2 to increase. Since the cost increased, EIGRP will no longer load balance across that path. The change in bandwidth and the effect on the routing table can be seen below:

In the above output notice that the traffic to 192.168.4.0/24 is no longer load balanced.

The maximum paths across which EIGRP can load balance can be configured using the **maximum-paths** *paths* command as shown below:

Unequal cost load balancing in EIGRP can be achieved using the **variance** command but is out of the scope of the CCNA exam.

One of the limitations that EIGRP inherits from distance vector protocols is the maximum hop count limitation. By default it has a maximum hop count of 100 but can be increased up to 255 using the **metric maximum-hops** *hops* command as shown below:

One important thing to remember when it comes to hop counts in EIGRP is that the count is not used in the calculation of cost of a route, but is only used to limit the size of an autonomous system.

## **5-6 Verifying and Troubleshooting EIGRP**

The following three commands are used to verify and troubleshoot EIGRP:

- show ip route
- show ip protocols
- show ip eigrp neighbors
- show ip eigrp topology
- debug eigrp packets and debug ip eigrp notifications

The **show ip route** command has been covered in the previous section and earlier in this chapter. Eventually a complete and correct routing table across the network is the best verification of a routing table. The other two commands are covered below.

### **Using show ip protocols command to verify and troubleshoot EIGRP**

The **show ip protocols** command helps verify routing protocols running on the router. An example of the output of this command from RouterA of our network is shown below:

The **show ip protocols** commands shows the operational information for EIGRP. From the above output you can gather that the EIGRP AS is 10 and it is advertising the 192.168.1.0 and 192.168.4.0 networks. You can also learn that it is using default metrics (K1 and K3) and the maximum hop count has been configured as 255. The output also shows that auto summary is disabled and maximum path is set to 6. As before, this output helps confirm the configuration of EIGRP.

### Using show ip eigrp neighbors command to verify adjacencies

It is important to see which routers EIGRP has formed adjacencies with and how stable the adjacencies are. The **show ip eigrp neighbors** command helps do this. The output from RouterE in our network is shown below:

The various fields in the output are discussed below:

- The H field indicates the order in which the neighbors were discovered.
- The Address indicates the IP address of the neighbor.
- The Interface indicates the interface via which the neighbor is reachable.

- The hold time shows how long this router will wait for a Hello packet to arrive from the neighbor.
- The uptime indicates the time since the adjacency was established.
- The SRTT or the smooth round-trip timer indicates the time it takes for a round-trip from this router to its neighbor and back. This value determines the amount of time router will wait after a multicast for a reply from this neighbor. If a reply isn't received in time, the router will switch to using unicasts.
- The RTO or Retransmission Time Out field indicates the amount of time EIGRP waits before retransmitting a packet from the retransmission queue to a neighbor.
- The Q value indicates whether there are any messages in the queue waiting to be sent to the particular neighbor. Consistently high values would indicate a problem in communication between the neighbors.
- The Seq, or sequence, field indicates the sequence number of the last update from the neighbor. As you know, EIGRP uses sequence numbers to keep track of updates and replies received to updates.

The various fields in the output help keep track of EIGRP adjacencies and their health. For example, high value in the SRTT, RTO or Q fields would indicate problems in communication.

### Using show ip eigrp topology command to verify and troubleshoot EIGRP

Since EIGRP stores all routes learned in a topology table, looking at the topology table can give you indications about the functioning of EIGRP as well as the stability of the network. The **show ip eigrp topology** command shows the topology table. An example from RouterC in our network is shown below:



Notice that a P precedes all routes. This means that the routes are in a **passive** state. A route in passive state means it is currently known about and stable. On the other hand, a route in **active** state (preceded by A) means that the route has lost the route and is searching for a replacement. Each route also shows the Feasible Distance (FD) and the next hop address towards the destination. The two numbers in the parenthesis are the feasible and advertised distances. For routes such as 192.168.4.0/24 notice that there are two next hops. The first one is the successor and will be seen in the routing table of the router while the second one is the feasible successor. The second route will be used if the first one is lost. If the feasible distance of two routes is displayed as the same, EIGRP will load balance between the two routes.

## Using debugs to troubleshoot EIGRP

As with RIP, there are some debugs for EIGRP and then help troubleshoot. The most important of all EIGRP debugs is the **debug eigrp packets**. Using this debug you can see EIGRP packets as they come into and go out of a router. On a router where EIGRP is working normally and has established adjacencies, you should only see hello packets coming in and going out in the debug output as shown below:

In the above output you can see hello packets being sent out every interface with the AS number shown. You can also see hello packets received from neighbors 192.168.2.2 and 192.168.6.5. The received packets also show the AS number. If the AS numbers in the updates do not match the AS number configured in the router, the packets will be dropped.

Another important debug for EIGRP is the **debug ip eigrp notifications**. During normal operations, you will not see any outputs for this debug but if there is change in the network. To generate the output below, I shutdown the s0/0 interface on RouterE, causing RouterC to loose the adjacency with that router.

When the s0/0 interface on RouterE is brought back up, the following debugs are seen on RouterC:

You can use this debug to verify that your network is stable and there are no constant changes in the network. No output for this debug command is good news!

## **5-7 Open Shortest Path First (OSPF)**

Open Shortest Path First (OSPF) is the first link-state protocol that you will learn about. Apart from being a link-state protocol, it is also an open standard protocol. What this means is that you can run OSPF in a network consisting of multivendor devices. You may have realized that you cannot run EIGRP in a network that consists of non-Cisco devices. This makes OSPF a very important protocol to learn.

Compared to EIGRP, OSPF is a more complex protocol and supports all features such as VLSM/CIDR and more. A brief summary of OSPF features is given below:


1. Works on the concept of Areas and Autonomous systems
2. Highly Scalable
3. Supports VLSM/CIDR and dis-contiguous networks
4. Does not have a hop count limit
5. Works in multivendor environment
6. Minimizes updates between neighbors.

While the above list is a very basic overview of the features of OSPF and will be expanded on in coming sections, it is a good time to take a step back and compare the four protocols detailed in this chapter. Table 5-2 shows a comparison of the four protocols.

**Table 5-2** *Comparison of routing protocols.*

Features	OSPF	EIGRP	RIPv1	RIPv2
Protocol Type	Link state	Hybrid	Distance Vector	Distance Vector
Classful Protocol	No	No	Yes	No
VLSM Support	Yes	Yes	No	Yes

<b>Discontiguous Network Support</b>	Yes	Yes	No	Yes
<b>Hop count limit</b>	None	255	15	15
<b>Routing Updates</b>	Event Triggered	Event Triggered	Periodic	Periodic
<b>Complete Routing table shared</b>	During new adjacencies	During new adjacencies	Periodic	Periodic
<b>Mechanism for sharing updates</b>	Multicast	Multicast and unicast	Multicast	Broadcast
<b>Best Path computation</b>	Dijkstra	DUAL	Bellman-Form	Bellman-Ford
<b>Metric used</b>	Bandwidth	Bandwidth and Delay (default)	Hop Count	Hop Count
<b>Organization type</b>	Hierarchical	Flat	Flat	Flat
<b>Convergence</b>	Fast	Very Fast	Slow	Slow
<b>Auto Summarization</b>	No	Yes	Yes	Yes
<b>Manual Summarization</b>	Yes	Yes	No	No
<b>Peer authentication</b>	Yes	Yes	Yes	No

It should be noted here that OSPF has many more features than the ones listed in Table 5-2 and than those covered in this [BOOK](#) . One feature that really separates OSPF from other protocols is its support of a hierarchical design. What this means is that you can divide a large internetwork into smaller internetworks called areas. It should be noted that these areas, though separate, still lie within a single OSPF autonomous system. This is distinctly different from the way EIGRP can be divided into multiple autonomous systems. While in EIGRP each autonomous system functions independent of others and a redistribution is required to share routes, in OSPF areas are dependent on each other and routes are shared between them without redistribution.

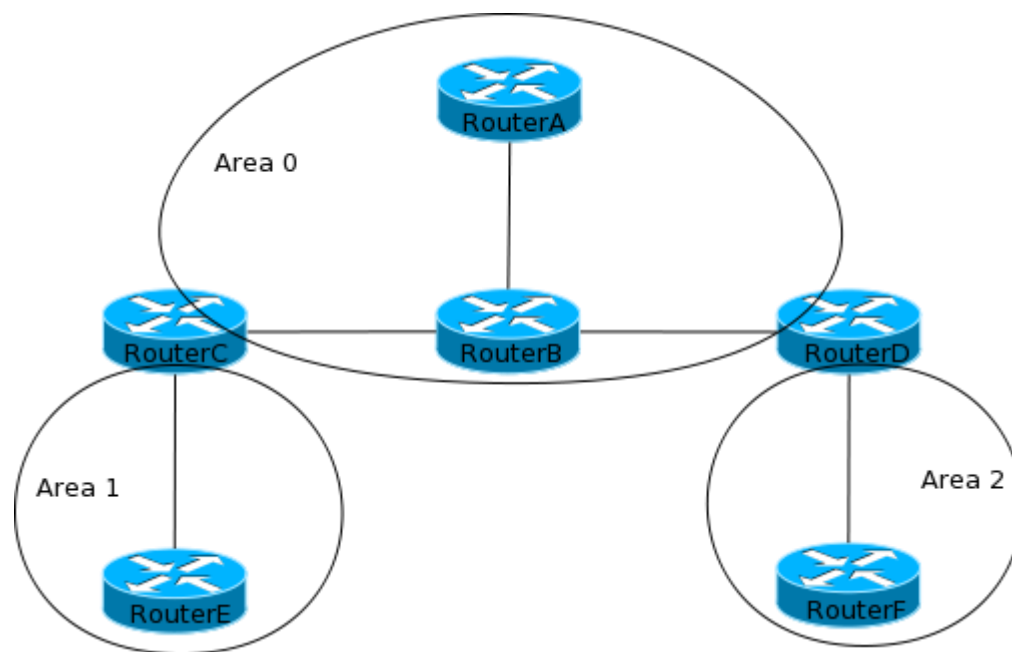
You should also know that like EIGRP, OSPF could be divided into multiple Autonomous Systems. Each autonomous system will be different from the rest and will require redistribution of routes.

The hierarchical design of OSPF provides the following benefits:

- Decrease routing overhead and flow of updates
- Limit network problems such as instability to an area
- Speed up convergence.

One disadvantage of this is that planning and configuring OSPF is more difficult than other protocols. Figure 5-5 shows a simple OSPF hierarchical setup. In the figure notice that Area 0 is the central area and the other two areas connect to it.

**Figure 5-5** *OSPF hierarchical design*



This is always true in an OSPF design. All areas need to connect to Area 0. Areas that cannot connect to area 0 physically need a logical connection it using something known as **virtual links**. Virtual links are out of the scope of the CCNA exam.

Another important thing to notice in the figure is that for each area, there is a router that connects to area 0 as well. These routers are called **Area Border Routers (ABRs)**. In Figure 5-5, RouterC and RouterD are ABRs because they connect to area 0 as well as another area. The way ABRs connect different areas, routers that connect different

autonomous systems are called Autonomous System Boundary Routers (ASBRs). In Figure 5-5, if RouterE connect to another OSPF AS or to an AS of another protocol such as EIGRP, it would be called an ASBR.

From Figure 5-5, you learned about three OSPF terms – Area, ABR and ASBR. Similarly there are many other terms associated with OSPF that you need to be aware of before getting into how OSPF actually works. The next section looks at some of these terms.

### Building Blocks of OSPF

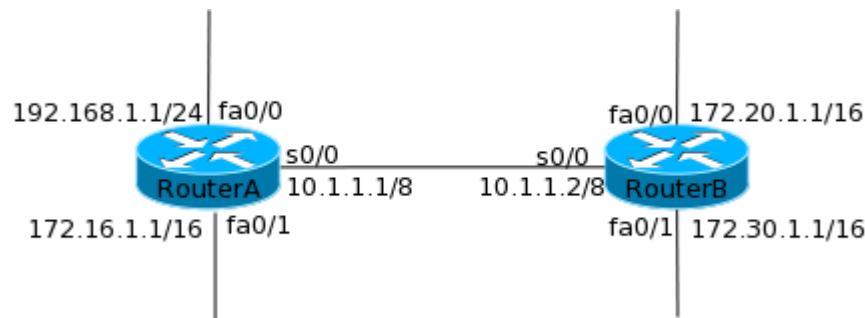
Each routing protocol has its own language and terminologies. In OSPF there are various terms that you should be aware of. This section looks at the some of the important terminologies associated with OSPF. In an attempt to make it easier to understand and remember, the terminologies are broken into three parts here – Router level, Area level and Internetwork level.

At the Router level, when OSPF is enabled, it becomes aware of the following first:

- **Router ID** – Router ID is the IP address that will represent the router throughout the OSPF AS. Since a router may have multiple IP addresses (for its multiple interfaces), Cisco routers choose the highest loopback interface IP address. (Do not worry if you do not know what loopback interfaces are. They are covered later in the chapter). If loopback interfaces are not present, OSPF chooses the highest physical IP address configured within the active interfaces. Here highest literally means higher in number (Class C will be higher than Class A because 192 is greater than 10).
- **Links** – Simply speaking a Link is a network to which a router interface belongs. When you define the networks that OSPF will advertise, it will match interface addresses that belong to those networks. Each interface that matches is called a link. Each link has a status (up or down) and an IP address associated with it.

Let's take a simple test here. Look at Figure 5-6 and try to find the Router ID and links on each of the routers.

Figure 5-6 RouterID and links



For RouterA, the RouterID will be 192.168.1.1 because it is the highest physical IP address present. The three links present on RouterA are the networks 192.168.1.0/24, 10.0.0.0/8 and 172.16.0.0/16. Similarly, the Router ID of RouterB is 172.30.1.1 since that is the highest physical IP address on the router. The three links present on RouterB are 10.0.0.0/8, 172.20.0.0/16 and 172.30.0.0/16.

Once a router is aware of the above two things, it will try to find more about its network by seeking out other OSPF speaking routers. At that stage the following terms come into use:

- **Hello Packets** – Similar to EIGRP hello packets, OSPF uses hello packets to discover neighbors and maintain relationships. Hello packet contains information such as area number that should match for a neighbor relation to be established. Hello packets are sent to multicast address 224.0.0.5.
- **Neighbors** – Neighbors is the term used to define two or more OSPF speaking routers connected to the same network and configured to be in the same OSPF area. Routers use hello packets to discover neighbors.
- **Neighbor Table** – OSPF will maintain a list of all neighbors from which hello packets have been received. For each neighbor various details such as RouterID and adjacency state are stored.
- **Area** – An OSPF area is a grouping of networks and routers. Every router in the area shares the same area id. Routers can belong to multiple areas; therefore, area id is linked to every interface. Routers will not exchange routing updates with routers belonging to different areas. Area 0 is called the **backbone area** and all other area must connect to it by having at least one router that belongs to both areas.



Once OSPF has discovered neighbors it will look at the network type on which it is working. OSPF classifies networks into the following types:

- **Broadcast (multi-access)** – Broadcast (multi-access) networks are those that allow multiple devices to access (or connect to) the same network and also provide ability to broadcast. You will remember that when a packet is destined to all devices in a network, it is termed as a broadcast. Ethernet is an example of a broadcast multi-access network.
- **Non-Broadcast multi-access (NBMA)** – Networks that allow multi-access but do not have broadcast ability are called NBMA networks. Frame Relay networks are usually NBMA.
- **Point-to-Point** – Point-to-Point networks consist of direct connection between two routers and provide a single path of communication. When routers are connected back-to-back using serial interfaces, a point-to-point network is created. Point-to-point networks can also exist logically across geographical locations using various WAN technologies such as Frame Relay and PPP.
- **Point-to-Multipoint** – Point-to-Multipoint networks consist of multiple connections between a single interface of a router and multiple remote routers. All routers belong to the same network but have to communicate via the central router, whose interface connects the remote routers.

Depending on the network type that OSPF discovers on the router interfaces, it will need to form **Adjacencies**. An **adjacency** is the relation between neighbors that allows direct exchange of routes. Unlike EIGRP, OSPF will not form adjacency with all neighbors always. A router will form adjacencies with a few or all neighbors depending on the network type that is discovered. Adjacencies in each network type is discussed below:

- **Broadcast (multi-access)** – Since multiple routers can connect to such networks, OSPF elects a **Designated Router (DR)** and a **Backup Designated Router (BDR)**. All routers in these networks, form adjacencies only with the DR and BDR. This also means that route updates are only shared between the routers and the DR and BDR. It is the duty of the DR to share routing updates with the rest of the routers in the network. If a DR loses connectivity to the network, the BDR will take its place. The election process is discussed later in the chapter.

- **NBMA** – Since NBMA is also a multi-access network, a DR and a BDR is elected and routers form adjacencies only with them. The problem with NBMA networks is that since broadcast capability and in turn multicast capability is not present, routers cannot discover neighbors. So NBMA networks require you to manually tell OSPF about the neighbors present in the network. Apart from this, OSPF functions as it does in a broadcast multi-access network.
- **Point-to-Point** – Since there are only two routers present in a point-to-point network, there is no need to elect a DR and BDR. Both routers form adjacency with each other and exchange routing updates. Neighbors are discovered automatically in these networks.
- **Point-to-multipoint** – Point-to-multipoint interfaces are treat as special point-to-point interfaces by OSPF and it does a little extra work on here that is out of scope of CCNA. There is no DR/BDR election in such networks and neighbors are automatically discovered.



**Exam Alert:** It can get confusing to remember the network types, election and adjacency requirements. A simple way to remember it is to associate “multi-access” with DR/BDR and “Point-to” with no election. Also associate NBMA with manually specifying neighbors.

Once OSPF has formed adjacencies, it will start exchanging routing updates. The following two terms come to use here:

- **Link State Advertisements** – Link State Advertisements (LSAs) are OSPF packets containing link-state and routing information. These are exchanged between routers that have formed adjacencies. The packets essentially tell routers in the networks about different networks (links) that are present and how to reach them. Different types of LSAs are discussed later in the chapter.
- **Topology Table** – The topology table contains information on every link the router learns about (via LSAs). The information in the topology table is used to compute the best path to remote networks.

At the area level, the only term that gets introduced is:

- **Area Border Routers (ABRs)** – Routers that connect an area to area 0 are called ABRs. They have one interface belonging to area 0 and other interfaces belonging to one or more areas. They are responsible for propagating routing updates between area 0 and other areas.

At the internetwork level another term that gets introduced is:

- **Autonomous System Boundary Router (ASBR)** – A router that connects an OSPF AS to another OSPF AS or AS belonging to other routing protocols is called an Autonomous System Boundary Router or ASBR. Route redistribution is setup between the two AS on these routers and hence they become the gateway between the two AS.

Now that you are familiar with OSPF terminology, the rest of the sections will discuss the working of OSPF in detail and help you better understand the terms discussed here.

## Loopback Interfaces

Loopback interfaces are virtual, logical interfaces that exist in the software only. They are used for administrative purposes such as providing a stable OSPF interface or diagnostics. Using loopback interfaces with OSPF has the following benefits:

- Provides an interface that is always active.
- Provides an OSPF Router ID that is predictable and always same. Making is easier to troubleshoot OSPF.
- Router ID is a differentiator in DR/BDR election. Having a loopback interface with higher order IP address can influence the election.

Configuring a loopback interface is easy – You need to select an interface number and enter the interface configuration mode using the **interface** command in global configuration mode as shown below:

The interface number can be any number starting from 0. Once in the interface configuration mode, use the **ip address** command to configure an IP address as you would on a physical interface. An example is shown below:

That's it! The loopback interface is configured and will be listed as an active interface in the **show ip interface** command.

The loopback interface can be important for OSPF because it will take the highest loopback IP address as the Router ID. If a loopback interface is not present, the highest physical IP address will be taken.

A loopback interface is logically equivalent to a physical address. The router is going to add an entry into its routing table for the network that the loopback interface address belongs to. So you can even configure a routing protocol to advertise the loopback network. Whether you choose to do that or not depends on whether you want the loopback address to be reachable from the network or not. Remember you will be using a subnet if you decide to advertise the loopback network.

### DR/BDR Election and influencing it

As discussed earlier, in multi-access network types, DRs and BDRs are elected and routers in the area only form adjacencies with them. So DRs and BDRs are an important part of OSPF and usually determine how well OSPF will function. In this section you will learn about the process by which DRs/BDRs are elected. Before learning about the process, it is important that you understand the terms **neighbors** and **adjacencies** fully since they are central to functioning of OSPF and the election process.

A router running OSPF will periodically send out Hello packets to multicast address 224.0.0.5. These hello packets serve as a way to discover neighbors. When a router

receives these packets, it checks the following to ascertain that a neighborhood can be established:

- **Area ID** – The Area ID received in a hello packet should match the area ID associated with the interface the packet was received on. As mentioned earlier, OSPF associates an area ID with each interface it is enabled on. The rationale behind comparing the area ID is that only router having interface in the same area should form neighborhood.
- **Hello and Dead intervals** – Hello packets exchanged by routers running OSPF contain information such as area ID, hello interval and dead interval. Hello interval specifies the time duration between hello packets and dead interval specifies the time duration after which a router will be declared dead if hello packets have not been received from it.

For a neighborhood to form, the hello and dead intervals should match between the routers.

- **Authentication** – OSPF allows you to set a password for an area. For neighborhood to form, the password must be same on the routers. Setting a password is optional.

If all the three above conditions match, the router will add the neighbor into the neighbor table and form a neighborhood. Even though a neighborhood gets formed, OSPF unlike EIGRP will not share routing updates, or link state advertisements in this case, with every neighbor.

For OSPF to share link state advertisements, an adjacency must be formed between the routers. As discussed earlier, how adjacencies are formed depends on the network type. In a multi-access network, a DR and BDR will be elected and all routers in the network will form adjacency with them only. Each router will exchange LSAs with DR and BDR. DR in turn will relay the information to the rest of the routers.

When routers realize that they are connected to a multi-access network, they will look at each Hello packet received to find the **priority** and **Router ID** of each router. Then the priority is compared and the router with the highest priority is selected the DR. The router with the second highest priority becomes the BDR. By default the priority of each router is 1 and can be changed on a per-interface basis.

If all routers have the default priority, then the router with the highest Router ID is elected the DR while the router with the second highest Router ID is elected the BDR. If the priority of a router is set to zero, it will not participate in the election process and will never be a DR or BDR.

As you know, the Router ID is the highest physical IP address present on a Router. This can be overridden by using a loopback interface because a router will use the highest loopback address, if one is present.

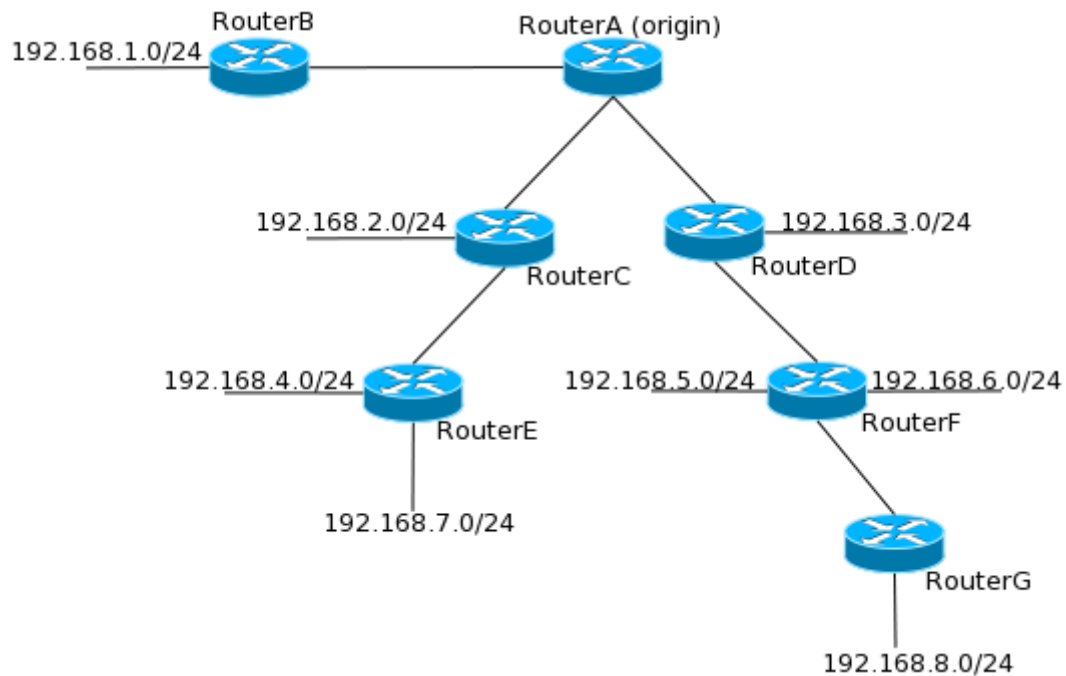
If you need to influence the DR/BDR election in a network segment, you can do one of the following:

- Manually increase the priority of a router interface to ensure that the router becomes the DR/BDR.
- Configure a loopback interface so that the Router ID becomes higher than that of the other routers in the network segment.

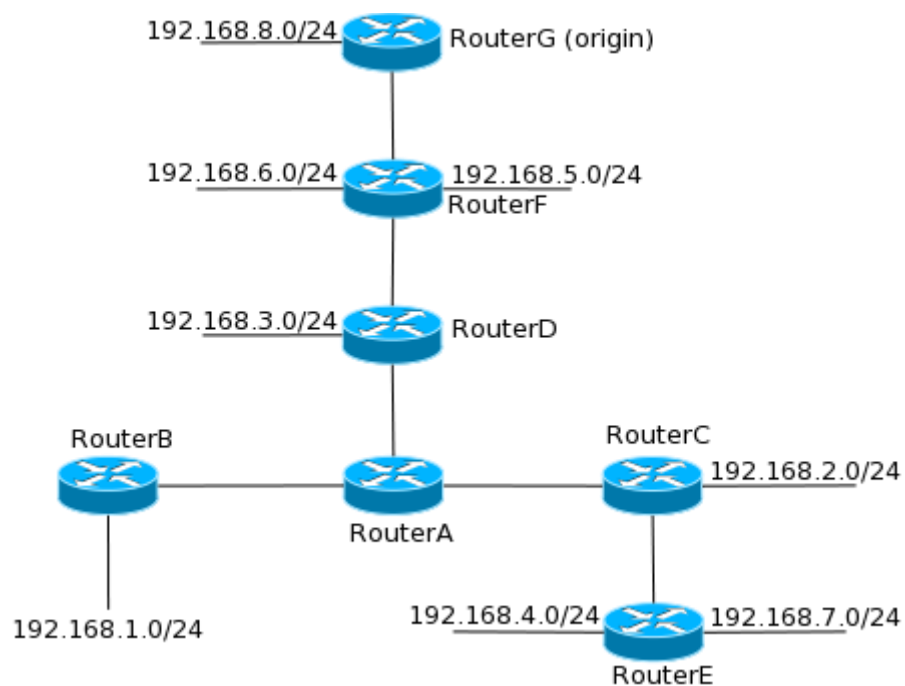
### SPF Tree Calculation

Once OSPF exchanges link state advertisements and populates the topology table, each router runs a calculation on the information collected. These calculations use something known as the **Shortest Path First** (SPF) algorithm. To do so, each router creates a tree putting itself at the root of the tree and the other routers and networks form the branch and leaves. In effect the router puts itself at the start and the area branches out from it. Figures 5-7 and 5-8 show an example of how the SPF tree is created by a router. Figure 5-7 shows the SPF tree with RouterA as the origin while Figure 5-8 shows the SPF tree with RouterG as the origin. Notice how different the network looks from the perspective of each router. The benefit of each router creating this tree is that the shortest path can be found from each router to each destination and there is no routing by rumor as seen with distance vector protocols.

**Figure 5-7** *SPF tree Example 1*



**Figure 5-8** *SPF tree Example 2*



It is important to understand that each router creates this tree only for the area it belongs to. If a router belongs to multiple areas, it will create a separate tree for each area.

A big part of the tree is also the **cost** associated with each path. Cost is the metric used by OSPF is the sum of the cost of the entire path from the router to the remote network. The OSPF RFC defines cost as an arbitrary value, so Cisco calculates cost as  $10^8/\text{bandwidth}$ . Bandwidth in this equation is the bandwidth configured on the interface. Using this equation, an Ethernet interface with a bandwidth of 10Mbps has a cost of 10 and a 100Mbps interface has a cost of 1. You may have noticed that interfaces having a bandwidth of more than 100Mbps will have a cost in fraction but Cisco does not use fractions and rounds of the value to 1 for such interfaces.

In Figure 5-8, if all interfaces are FastEthernet interfaces with a bandwidth of 100Mbps, each link has a cost of 1. So for the path from RouterG to the 192.168.7.0/24, the total cost will be 5 and to the network 192.168.3.0/24, the total cost will be 2.

The cost of each interface can be changed using the **ip ospf cost** command in the interface configuration mode. It should be noted that since the OSPF RFC does not exactly define the metric that makes up the cost, each vendor uses a different metric. When using OSPF in a multivendor environment, you will need to adjust cost to ensure parity.

### Link State Advertisements

The fundamental building blocks of OSPF are the link state advertisements that are sent from every router to advertise links and their states. Given the complexity and scalability of OSPF, different LSA types are used to keep the OSPF database updated. Out of the various LSAs, the first five are most relevant to the limited OSPF discussion covered in this chapter and are discussed below:

- **Type 1 – Router LSA** – Each router in the area sends this LSA to announce its presence and list the links to other routers and networks along with metrics to them. These LSAs do not cross the boundary of an area.
- **Type 2 – Network LSA** – The DR in a multi-access network sends out this LSA. It contains a list of routers that are present in the network segment. These LSAs also do not cross the boundary of an area.



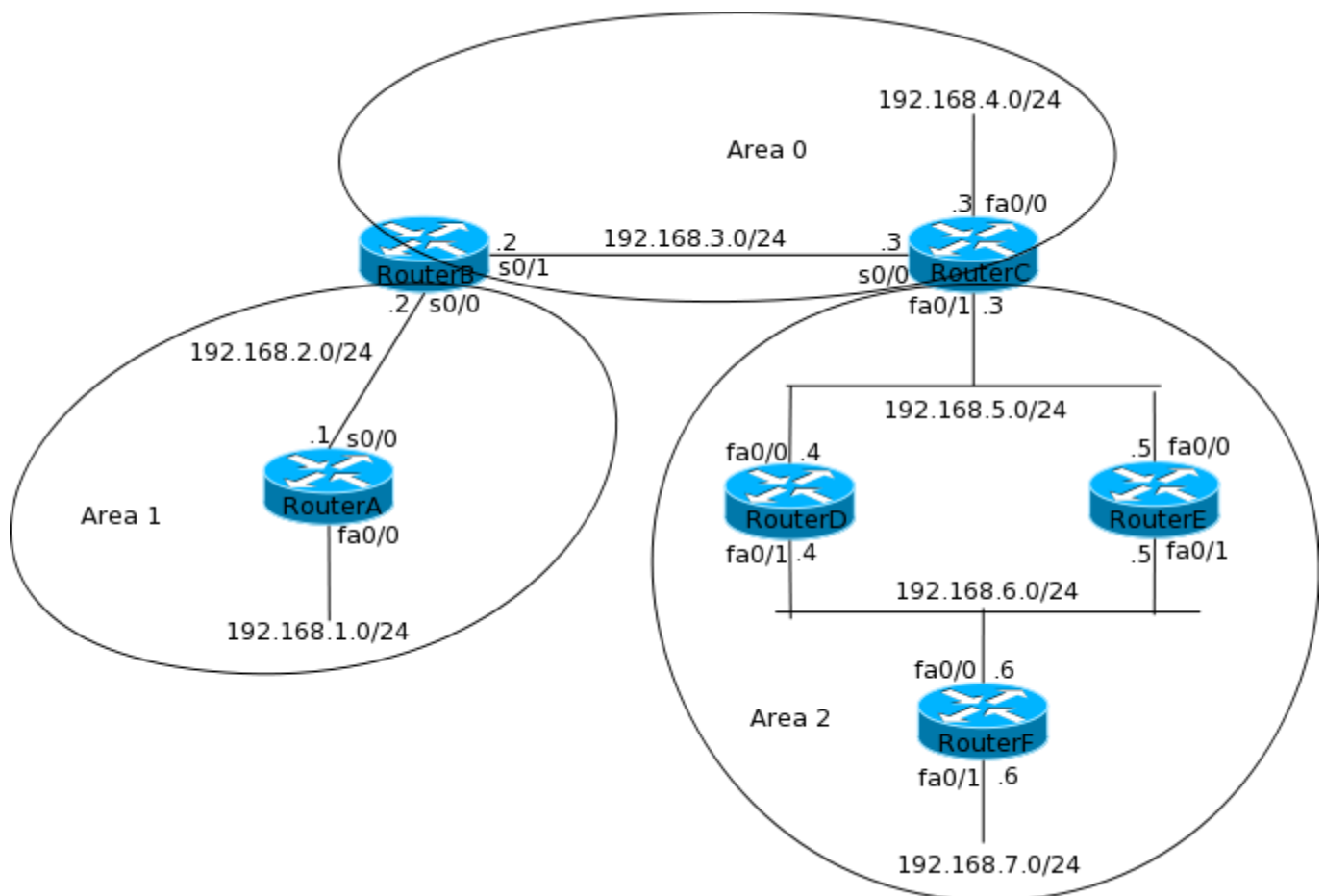
- **Type 3 – Summary LSA** – The ABR takes the information learned in one area (and optionally summarizes this information) and sends it out to another area it is attached to. This information is contained in LSA type 3 and is responsible for propagation of Inter-area routes.
- **Type 4 – ASBR Summary LSA** – ASBRs originate external routes (redistributed routes) and send them throughout the network. While the external routes are listed in type 5 LSA, the details of the ASBR themselves are listed in type 4 LSAs. This LSA is originated by the ABR of the area where the ASBR resides.
- **Type 5 – External LSA** – This LSA lists routes redistributed into OSPF from another OSPF process or another routing protocol. This LSA is originated by the ASBR and propagates across the OSPF AS.

## 5-8 Configuring OSPF

### Configuring OSPF

In the previous section you learned about OSPF and how it works. While a lot of theory was covered in that section, this one looks at configuring OSPF. The network shown in Figure 5-9 will be used for this section.

**Figure 5-9** OSPF Network



Just like EIGRP, OSPF configuration is divided into two parts – the global configuration and the interface level configuration. Globally, configuring OSPF includes enabling the process and adding networks to be advertised. To enable the OSPF process use the **router ospf process\_id** global configuration command. In this

command, *process\_id* is a locally significant number and does not represent the AS. Since multiple OSPF processes can run on a router, the *process\_id* is used to keep the processes separate. The process id can be different on every router. On entering the command, you will arrive at the router configuration mode when the **network** command can be used to specify the networks that will be advertised. In respect to OSPF, the network command actually identifies the interfaces on which OSPF will be enabled and the network to which the interface belongs will be advertised. The syntax of the command is:

In the above command, *network\_number* and *wildcard\_mask* combine to identify the interface on which OSPF will be enabled. While *network\_number* is a network address (example 192.168.1.0), a *wildcard\_mask* is the inverse of subnet mask. In a subnet mask a 255 is an octet means the corresponding octet in the network number should match exactly while a 0 means it can be anything. In a wildcard mask a 255 means the corresponding octet in a network number can be anything and 0 means the corresponding octet should match exactly. Confusing? Table 5-3 shows a few examples. The last component of the above command is the *area\_id* to which the interface will belong.

**Table 5-3** Wildcard Mask examples

Network	Wildcard	Matches	Explanation
<b>192.168.1.1</b>	0.0.0.0	192.168.1.1	Since each octet of the wildcard mask is a zero, only an interface with an IP address specified by the network number will be matched.
<b>192.168.1.0</b>	0.0.0.255	192.168.1.0- 192.168.1.255	Since the last octet is 255, the value of the last octet of the network number does not matter as long as the first three octets match the given network number.
<b>10.1.0.0</b>	0.255.255.255	10.0.0.0- 10.255.255.255	A wildcard mask of 0.255.255.255 means that as long as the first octet matches, the rest can be anything.

If you want to enable OSPF on each interface individually then you can use a wildcard mask of 0.0.0.0 with network numbers consisting of the IP address of each interface. This is the simplest and easiest way to configure OSPF but you can also use wildcard mask to cover a range of addresses. For example, In Figure 5-9, RouterF has two interfaces and OSPF can be enabled on them using two **network** commands like:

```
network 192.168.6.0 0.0.0.255 area 2
network 192.168.7.0 0.0.0.255 area 2
```

Another way to configure OSPF on RouterF is to use a single network commands as shown below:

While the first method is precise and safe, the second method can introduce problems since it covers a wide range of networks. Another method to use wildcard masks is to specify network blocks. Wildcard masks can represent blocks of network just like network masks. You may recall that network masks can represent only specific sizes of blocks – 2, 4, 8, 16, 32, 64, 128 and 255. To specify a block with wildcard mask simply deduct one from the block size. For example, in RouterF networks 192.168.6.0 and 192.168.7.0 can be covered using any of the following:

Now that you understand the **network** command, let us configure the OSPF in the network shown in Figure 5-9. I will be using a mix of wildcard masks types discussed above to configure each router. Make sure you pay attention to the area each interface should belong to. With OSPF, area 0 should be configured first, so we will start with routers belonging to area 0.

### Area 0

RouterB has one interface in area 0 while RouterC has 2 interfaces in area 0. The best way to configure them is to use a wildcard mask of 0.0.0.0 with the interfaces addresses.

Note that 192.168.3.0 and 192.168.4.0 cannot be configured as a single block since a block of 2 or 4 will not cover them and a block of 8 will cover 192.168.5.0, which is in area 2. So we had to use two statements with a mask of 0.0.0.0.

### Area 1

Now that area 0 has been configured, other areas can be configured. RouterA has two interfaces in area 1 and RouterB has one interface in that area.

Notice that on RouterA a network number of 0.0.0.0 and a wildcard mask of 255.255.255.255 are used. This mask essentially means all networks and can be used on RouterA since both the interfaces belong to area 1.

## Area 2

The final area spans across four routers. All interfaces of RouterD, RouterE and RouterF belong to area 2.

In the above configuration notice the three different ways wildcard has been used on RouterD, RouterE and RouterF.

Now that OSPF configuration is complete, let us take a look at the routing table on each router to verify the configuration.







The above outputs show that all networks are known across the internetwork. You should also notice the following:

- While an O precedes OSPF routes, inter-area routes are preceded with an IA also.
- In the outputs from RouterC and RouterF notice that OSPF is load balancing across equal cost paths.

### Influencing path selection

As discussed in the earlier section, Cisco uses interface bandwidth as a metric for cost and the sum of cost of the entire path is used to select the best route to a destination. The cost of an interface can be manually changed using the **ip ospf cost** command in the interface configuration mode.

For example, in the network shown in Figure 5-9, traffic going from RouterF to 192.168.4.0/24 is being load balanced between the paths going through RouterD and RouterE. RouterF can be made to route traffic only through RouterD and use RouterE as a backup path by increasing the cost associated with interface fa0/0 on RouterE. This will cause the cost of the entire path to increase causing RouterF to not use that path along with the other path. The following commands will increase the cost on RouterE:

The effect of this change will almost immediately be seen on the routing table of RouterF:

In the output above, notice that RouterF is no longer load balancing the traffic across the two paths.

### **Influencing DR/BDR election**

In the previous section you learned that OSPF routes do not form adjacencies with all neighbors in a multi-access network. A DR and a BDR are elected and all other routers form adjacencies with them. This election takes into consideration the OSPF priority and in case of a tie, the Router ID.

For example, in the network shown in Figure 5-9, RouterE will be the DR and RouterD will be the BDR in the Ethernet network 192.168.5.0/24 because RouterE has the highest router ID (192.168.6.5) and RouterD has the second highest router ID (192.168.6.4). RouterC has a router ID of 192.168.6.3. If you wanted RouterC to always

be the DR, either the priority or the Router ID would have to be increased. The easiest way to do this is to increase the priority on interface fa0/1 of RouterC as shown below:

The **show ip ospf interface** *interface* command can be used as shown below:

While the entire output is discussed in the next section, notice that the third line shows the state as DOTHER. DOTHER means that the router is not a DR or BDR. So why did the new priority not cause RouterC to become the DR or BDR?

That is because a DR/BDR election does not take place till the existing DR/BDR leaves the network. One way to force a reelection is to restart the OSPF process on the current DR and BDR. A reset will cause the OSPF process to restart and the network will think that the DR and BDR are lost and will force an election. You can reset the process using **clear ip ospf process** command. To force an election in the network, let's reset the OSPF process on RouterD and RouterE as shown below:

Once the adjacencies are reestablished, the output of **show ip ospf interface** on RouterC looks like the following:

Notice that RouterC is now the DR. Another way to influence the election would have been to create a loop back interface on RouterC with a high IP address as shown below:

This would cause the router ID of RouterC to be higher than the rest.

## **5-9 Verifying and Troubleshooting OSPF**

There are various ways to verify and troubleshooting OSPF configuration and operation. The following are the most useful:

1. show ip protocols
2. show ip ospf
3. show ip ospf interface
4. show ip ospf neighbor
5. show ip ospf database
6. debug ip ospf packet
7. debug ip ospf hello
8. debug ip ospf adj

### **Using show ip protocols command to verify and troubleshoot OSPF**

As with other routing protocols, **show ip protocols** helps verify the global configuration of OSPF. The output of this protocol from RouterA in our network is shown below:

The first thing to notice in the above output is the router ID and the areas configured on this router. It also shows the networks that are added to the process. The final thing to notice is that the adjacent router is shown as the Routing Information Source. As you can see, the output from this command can be used to quickly verify the basic configuration and it is easy to catch any configuration mistake in this small output.

### Using show ip ospf command to verify and troubleshoot OSPF

The **show ip ospf** command is also useful to verify configuration. While most of the output is out of scope of CCNA, a few things such as Router ID, Area related information, and SPF related information is useful. The output from RouterA is shown below:

In the above output, notice that SPF algorithm was run twice. This means there was a change in the network once after OSPF started.

### Using **show ip ospf interface** command to verify and troubleshoot OSPF

One of the most important commands used to verify and troubleshoot OSPF is the **show ip ospf interface** command. It can be used to see information of all interfaces



participating in OSPF or any specific interface. A sample output from RouterD is shown below:

In the output of the **show ip ospf interface** command, you will get a lot of information regarding the interfaces participating in OSPF as well as the network segment each interface is attached to. The following information is of particular importance:

1. The Interface IP
2. Area that the interface belongs to
3. The Router ID
4. Network type
5. The DR and BDR IP address in that segment (in case of multi-access networks)
6. The hello and dead timers
7. Neighbors and Adjacency count
8. Reason of adjacency (in case of multi-access networks)

In the above output, notice that RouterD is not the DR or BDR for both the network segments it is connected to.

### Using show ip ospf neighbor command to verify and troubleshoot OSPF

One of the most important parts of OSPF is the neighborhood discovery and formation of adjacency. Hence the output from the **show ip ospf neighbor** command is very important. There are some variations in the output of this command and the best example is the output from RouterC because it consists of multiple types of interfaces. The output from RouterC is shown below:

The fields of the above output are discussed below:

1. **Neighbor ID** – This field lists the Router ID of all the neighbors discovered. Remember this is the Router ID of the neighbor and not the IP address of the interface by which the router is connected to this router.
2. **Pri (Priority)** – This is the interface priority of the neighbor. A priority of 0 is seen if the neighbor is manually configured with this priority or if the network between then neighbors is not a multi-access network.
3. **State** – This field indicates the state of adjacency. FULL indicates that a complete adjacency has been established with the neighbor and database has been exchanged. The second value indicates the type of adjacency. Possible values are DR (the adjacent router is a DR), BDR (the adjacent router is BDR), DOTHER (the adjacent router is a router in the network) and – which means that it's an adjacency across a non-multi-access link. If the first value is stuck as 2WAY, this means that an adjacency has not been formed. Possible reason for this is that the neighbor is not a DR or BDR in a multi-access network. An adjacency stuck at any other state is reason for concern.
4. **Dead Time** – This field indicates the period after which the neighbor will be declared dead if a hello packet is not received.
5. **Address** – This field indicates the interface IP address of the neighbor that is connected to the same network as this router.
6. **Interface** – The interface towards which the neighbor can be reached.

**Using show ip ospf database command to verify and troubleshoot OSPF**

The **show ip ospf database** command shows two important things regarding the area a router is connect to – the number of routers in the area and the network links known in the area. The output is broken down by area. An example of the output from RouterC is given below:

You can see that routers available in area 0 and area 2 are shown along with links known in the area. ADV Router stands for advertising router and the field shows the router ID of the originating router for each link. Inter-area links will show as originating from the ABR.

### Using debug ip ospf packet to verify and troubleshoot OSPF

Like EIGRP, OSPF packets can be seen entering and leaving a router. The command to see the packets is **debug ip ospf packet**. Ideally in a stable network you will only see the hello packets as shown below in the output from RouterC:

The above output shows the hello packets received from the three neighbors on RouterC.

### Using debug ip ospf hello to verify and troubleshoot OSPF

While the **debug ip ospf packet** command shows all packets, the **show ip ospf hello** command can be used to specifically look at hello messages sent and received on a router. This can be useful to troubleshoot neighborship and adjacency problems. Output from RouterC is shown below:

In the above output you can see that RouterC is sending hello packets out all its interfaces and is receiving hello packets back from all neighbors. If there is a problem with hello packets such as an interval mismatch, the debug will show that error.

### Using **debug ip ospf adj** to verify and troubleshoot OSPF

As mentioned earlier, adjacency formation is the most important part of OSPF operation and most problems occur at that stage. The output from **debug ip ospf adj** helps identify problems related to an adjacency. Since there are no adjacency related events in a stable network, I cleared the ospf process on RouterB to generate the following output on RouterC:

The above output shows the transition of the adjacency through various stages – 2WAY, EXSTART, EXCHANGE and finally FULL. While these states are out of scope of

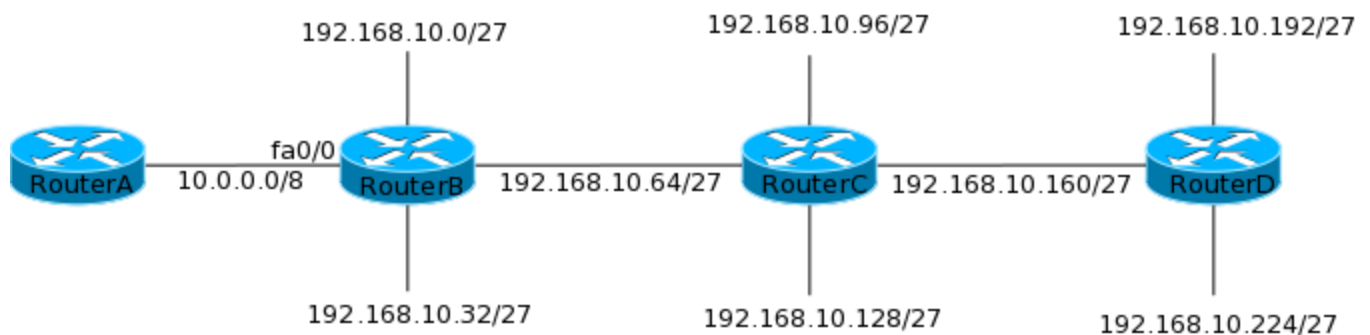
CNA, the output gives you an idea of how an adjacency is formed. In the EXCHANGE state, the database is synchronized and then the FULL state is reached. Any problems during any of these states will be seen in this debug.



### **5-10 EIGRP and OSPF Summary & Redistribution Routes**

As you know from Chapter 2, summary routes can be used to group together various contiguous networks into a single route. This is useful for reducing the size of routing table in the network. For example, in the network shown in Figure 5-10, if summarization is disabled and all routers are running EIGRP in the same AS, RouterA will have 8 EIGRP routes in its routing table.

**Figure 5-10** EIGRP Summarization



The routing table of RouterA, with summarization disabled is shown below:

You may have noticed that all these 8 networks are contiguous networks and can be summarized into a single 192.168.10.0/24 route. In this section you will learn to configure summarization on EIGRP and OSPF.

When configuring summarization on EIGRP, remember that by default EIGRP summarizes on network boundaries. In the above shown network EIGRP would have summarized the 192.168.10.x network when advertising the routes from RouterB to RouterA because 10.0.0.0/8 network falls between them. Before configuring manual summarization, you should disable automatic summarization using the **no auto-summary** command in under the routing protocol configuration as shown below:

Summarization is configured on a per-interface basis. EIGRP will summarize the routes when advertising out the interface. The following command is used to configure summarization on the interface:

While the above command is self explanatory, particular emphasis must be given to the subnet-mask. The subnet mask defines the block size you want to summarize into. For example, we can summarize the network shown in Figure 5-10 into a single /24 route using the following command:

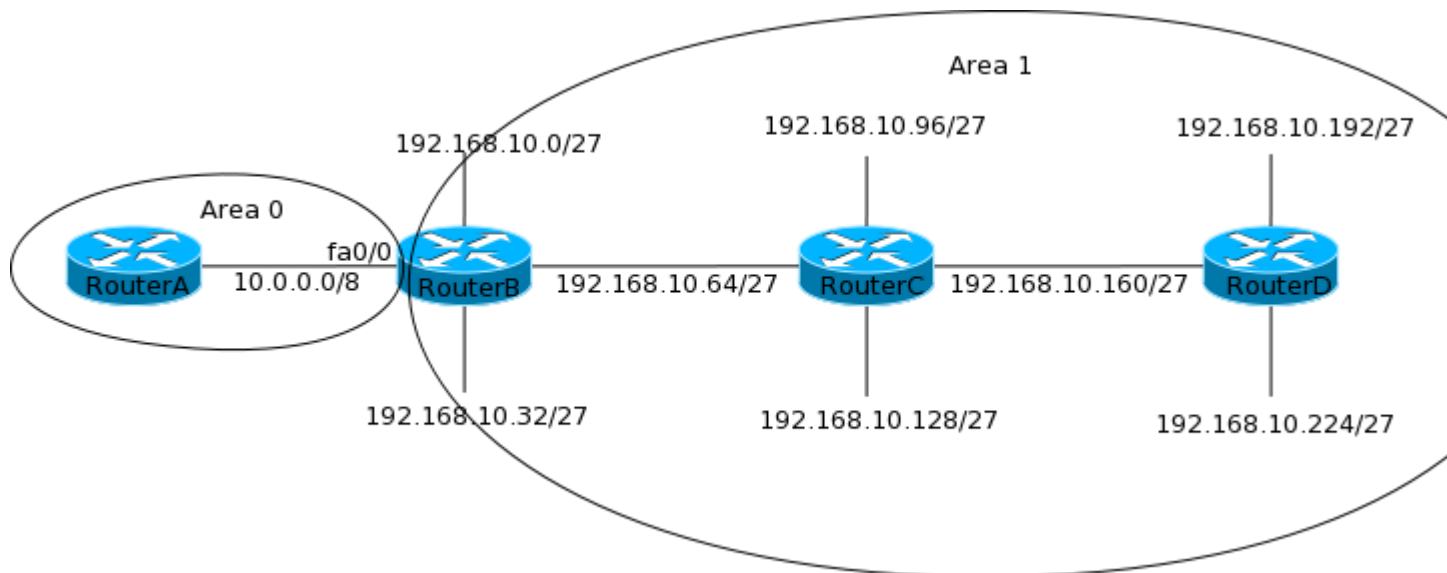
After the above command, the routing table on RouterA will look as shown below:

We can also summarize the networks into two /25 networks using the following commands:

After the above commands, the routing table on RouterA will look as shown below:

If the network shown in Figure 5-10 was running OSPF and was divided into areas as shown in Figure 5-11, you can configure area 1 to send a summary route to area 0.

**Figure 5-11** OSPF Summarization



Remember that only an ABR can summarize a route, so you will need to configure summarization on RouterB using the following command in the OSPF configuration:

The network-address and subnet-mask values function the same way as in EIGRP. In our example, you will need to add the following commands on RouterB to summarize all the 8 subnets into a single /24 summary route:

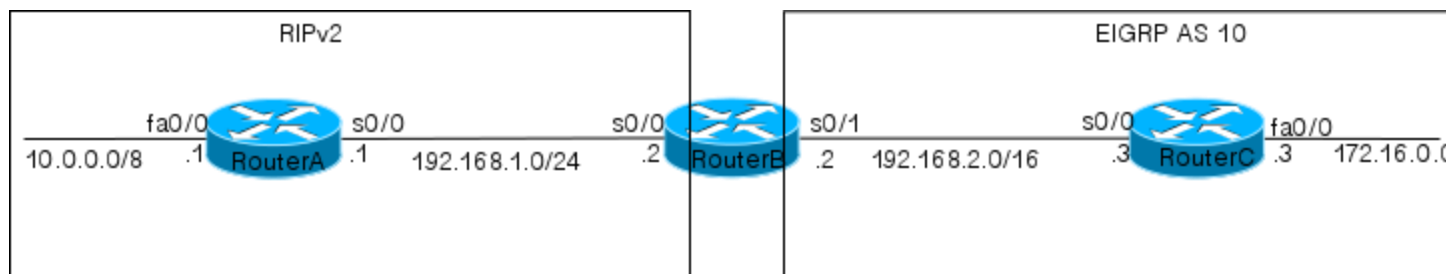
Since OSPF does not summarize automatically, you do not need the **no auto-summary** command here.

## Redistributing Routes

In Chapter 4, you were introduced to redistribution. The CCNA exam requires you to have a keen knowledge of redistribution. In particular you are required to know how to redistribute routes in RIP and this sections looks at that. For this section, the network shown in Figure 5-12 will be used.

In the network shown in Figure 5-12, RIPv2 is running on RouterA and RouterB while EIGRP is running on RouterB and RouterC. RouterA has no route towards 172.16.0.0/16 network that is being advertised by EIGRP.

**Figure 5-12** *Redistributing Routes*



As you know from Chapter 14, while redistributing routes into a protocol, the metric compatibility much be ensured. In this case, routes to 172.16.0.0/16 will have EIGRP metrics and those tend to be large numbers. On the other hand, anything above 15 is an invalid metric for RIP. To overcome this, RIP must be told what metric to assign to the routes redistributed from EIGRP. To redistribute the routes, the **redistribute protocol [process-id] metric metric** command is used in the routing protocol configuration mode. To redistribute EIGRP routes into RIP in the given network, the following commands are required on RouterB:

The above command will cause RouterB to redistribute routes to 192.168.2.0/16 and 172.16.0.0/16 networks into RIP. RIP in turn will advertise these routes to RouterA with a metric of 2. The routing table of RouterA, after redistribution will look as shown below:

In the above output notice that both the routes are learned from RIP and have a metric of 2.

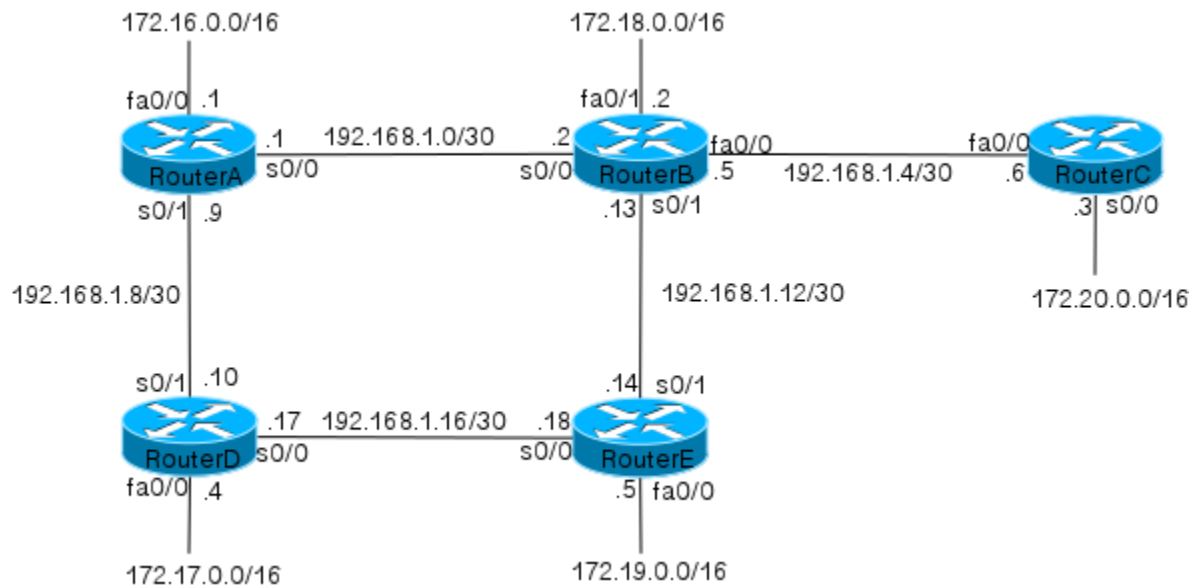
## **5-11 Lab 5-1: RIP**

You have been tasked with configuring the network shown in Figure 5-13 using RIPv2 such that:

1. All interfaces on each router are advertised in RIP
2. RouterC should not learn routes from the rest of the network. It should use a default route to reach remote networks. All routers should learn the 172.20.0.0/16 network using RIP
3. All interfaces that do not connect to another router such not advertise RIP routes.
4. Remember that the DCE side of your DTE/DCE back to back cable should be connected to the interface configured with clock rate.

**Figure 5-13 Network Setup for Lab 5-1**

## TUN MIN OO {BE-IT} Routing & Switching 200-120



The initial configuration for each router is given below

### RouterA

### RouterB



RouterC

**RouterD**

**RouterE**

## Solution

First, each interface on each router needs to be added in RIP and version 2 has to be enabled:

The second item in the list states that RouterC should not learn any routes from the rest of the network, while the rest of the network should learn routes originated by it.

RouterC also needs to have a default route to the rest of the network. To achieve this, RouterB's f0/0 interface must be made passive so that it does not advertise the routes out this interface to RouterC while it still learns the routes advertised by RouterC. The configuration required is shown below:

The final item in the list states that routes should not be advertised out interfaces that do not connect to another router. This requires some interfaces on all routers to be passive:

## Verification

To verify the solution, first check the routing table on each router. The routing table should resemble the output shown below:



In the above outputs notice that RouterC does not have any RIP routes but all other routers know network 172.20.0.0/16.

A final verification can be done by sending a ping to 172.20.0.3 (interface s0/0 of RouterC) from RouterD as shown below:

A successful ping shown that routing is working perfectly in the network.

## **5-12 Lab 5-2: EIGRP**

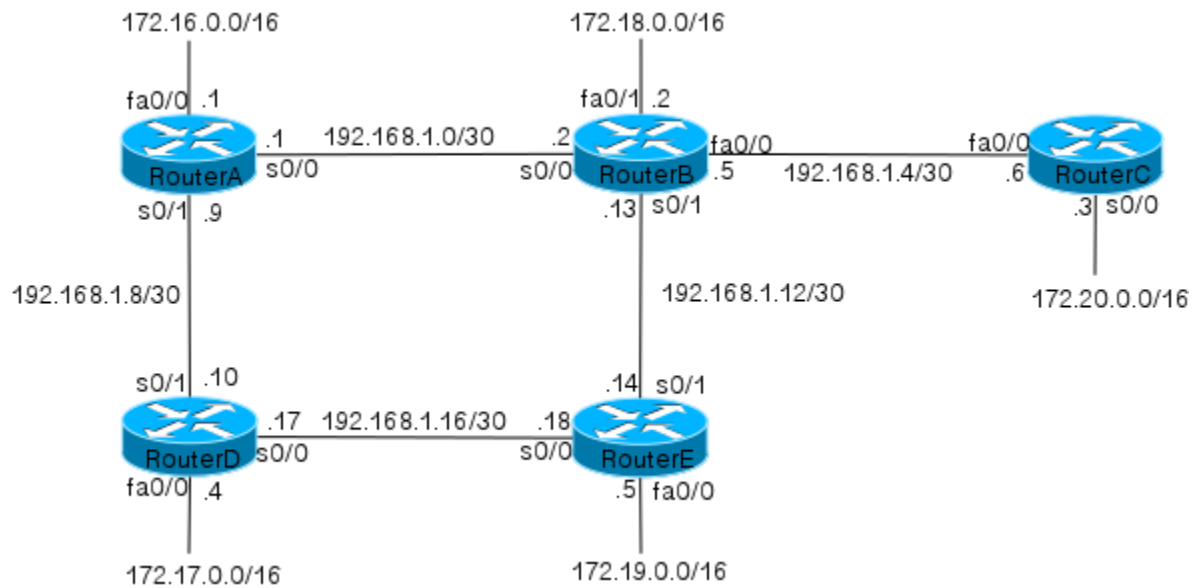
You are tasked with configuring the network shown in Figure 5-14 using EIGRP such that:

1. Each interface on every router is advertised in EIGRP AS 1
2. Traffic from 172.17.0.0/16 network destined to 172.20.0.0/16 should take the RouterD->RouterA->RouterB->RouterC path. If that path is not accessible, the traffic should be routed via RouterE.
3. Ensure that EIGRP can support dis-contiguous networks.
4. RouterD should have only a summary route for all 192.168.1.x networks not directly connected to it.
5. Remember that the DCE side of your DTE/DCE back to back cable should be connected to the interface configured with clock rate.

**Figure 5-14 Network for Lab 5-2**



## TUN MIN OO {BE-IT} Routing & Switching 200-120



The initial configuration for all routers is shown below:

### RouterA

### RouterB

RouterC

**RouterD**

**RouterE**

## Solution

First, each interface on each router needs to be added in RIP and version 2 has to be enabled:

## Solution

The first item requires configuring EIGRP on all routers and advertising all interfaces as shown below:

The second item in the list requires that traffic is not load balanced from RouterD to RouterA and RouterE. Both paths have an equal cost, so the metrics must be modified to stop EIGRP from load balancing as shown below:

The next item in the list requires auto summarization to be disabled on all routers as shown below:

The final item requires you to configure summarization on RouterA and RouterE. Since you can only configure summarization in blocks, you will need to summarize the addresses in block of 32 to cover all 192.168.x.x/30 networks as shown below







In the routing table of RouterD, notice that it has only a single route to 172.20.0.0/16 network whose next hop is RouterA. Also notice that there is a summary route on it for all the 192.168.1.x networks.

Finally, ping the 172.20.0.3 network from RouterD to verify that routing is working correctly:

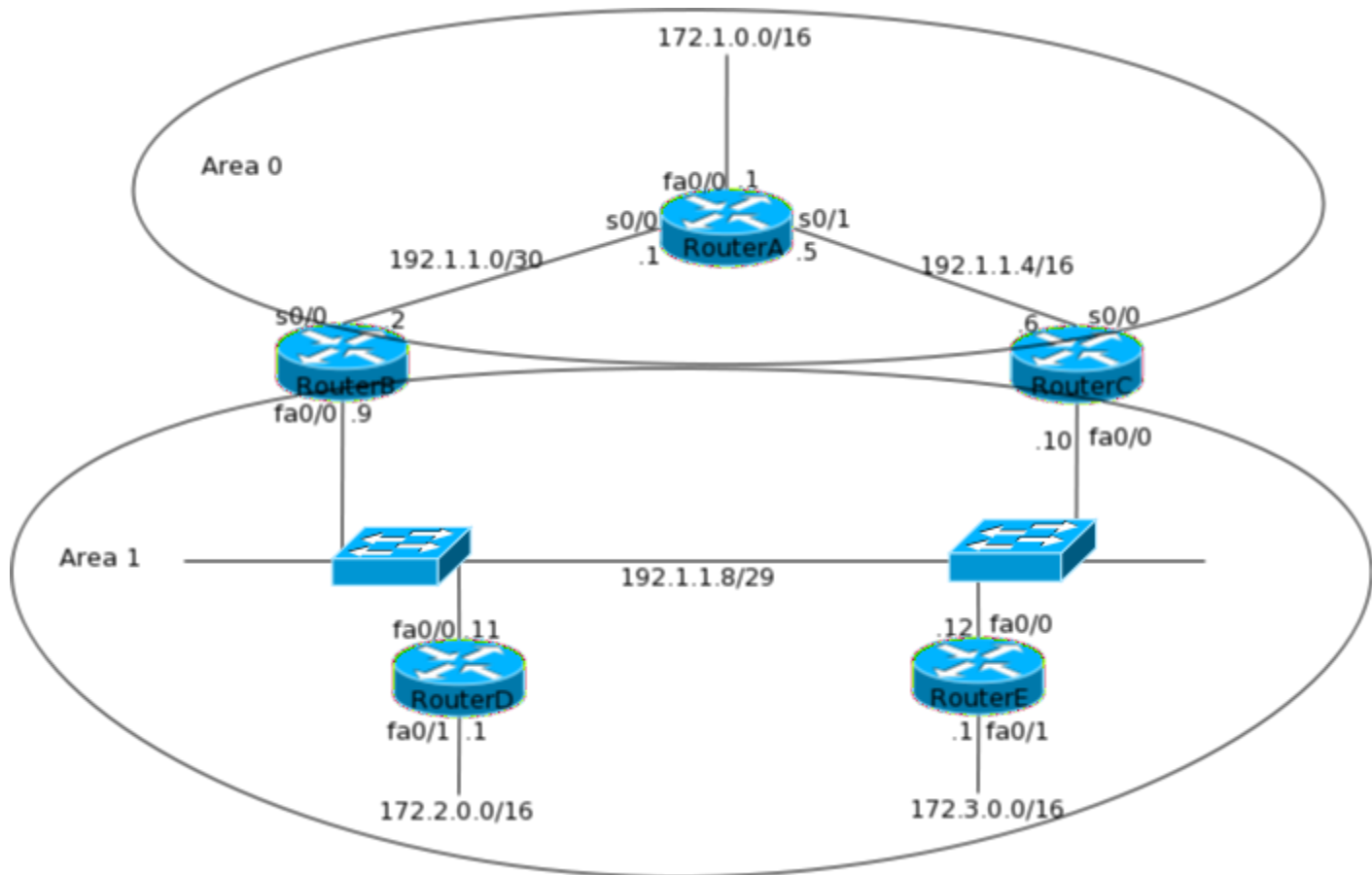
## **5-13 Lab 5-3: OSPF**

You are tasked with configuring the network shown in Figure 5-15 with OSPF such that:

1. All interfaces on every router is added in OSPF.
2. Router IDs for each router should be:
  1. RouterA – 1.1.1.1
  2. RouterB – 2.2.2.2
  3. RouterC – 3.3.3.3
  4. RouterD – 4.4.4.4
  5. RouterE – 5.5.5.5
3. Any networks created for item 2 on this list should not be advertised in OSPF
4. Router E never becomes the DR/BDR in the 192.1.1.8/29 network segment
5. Router D is always the DR in in the 192.1.1.8/28 network segment

6. RouterE should use RouterC to reach 172.1.1.0/16 network. The other path should be used for backup only.
7. Remember that the DCE side of your DTE/DCE back to back cable should be connected to the interface configured with clock rate.

Figure 5-15 Network for Lab 5-3



The initial configuration of the routers is shown below:

#### RouterA

**RouterB**

**RouterC**

RouterD

RouterE

## Solution

Before configuring OSPF globally, configure RouterE with an OSPF priority of 0 and RouterD with a priority of 10 to complete item 4 and 5 in the list. Configuring these before adding networks in OSPF will ensure that DR/BDR is influence at the first go without having to restart OSPF process later.

Next, configure a loopback interface on each router to change the Router ID as given in item 1 in the list:

Next, configure OSPF on all routers as shown below. Remember not to use 0.0.0.0 255.255.255.255 to advertise networks since loopback interfaces should not be advertised as per item number 3 in the list.



Since RouterE has two equal cost paths to reach the 172.1.0.0/16 network, it will load balance across them. To disable load balancing, the cost needs to be changed as shown below:

### Verification

To verify the solution, first take a look at the routing table of each router in the network:



In the above output notice the following:

1. All networks are seen across the internetwork
2. Loopback networks are not advertised across the network
3. RouterE has a route towards 172.1.0.0/16 network through 192.1.1.10 (RouterC)

Next, verify the DR/BDR election in the Ethernet segment by looking at the neighbor table on RouterE:


```
RouterE#sh ip ospf neighbor
```

```
Neighbor ID Pri State Dead Time Address Interface
2.2.2.2 1 2WAY/DROTHER 00:00:39 192.1.1.9 FastEthernet0/0
3.3.3.3 1 FULL/BDR 00:00:31 192.1.1.10 FastEthernet0/0
4.4.4.4 10 FULL/DR 00:00:31 192.1.1.11 FastEthernet0/0
```

Notice that the RouterID of every router is as listed in item 2 of the list and that the RouterD is the DR.

Finally, ping the 172.1.0.0/16 network from RouterE to confirm that routing is working properly:

## Summary

Phew! This was a big chapter and with a good reason too! As I mentioned earlier in the [BOOK](#) , the CCNA certification is mostly about the network and the data-link layer. This chapter covered the most important aspect of the network layer – routing protocols. Here you got to know what binds a network together.

In this chapter you were introduced to all three routing protocol – RIP, EIGRP and OSPF. You learned the how each one operates and how to configure them. You also learned the difference between how each of these protocol works. I cannot stress enough the importance of this chapter. I strongly suggest re-reading the chapter and

practicing configuring, verifying and troubleshooting them before moving to the next chapter because the next two chapters look at the data-link layer that is very different from the network layer.

## **Chapter 6 Switching and Spanning Tree Protocol**

In the first chapter, you were introduced to bit of switching and switches. You already know that the layer 2 of the OSI model deals with switching frames in the local network and that switches work at this layer. You also know that switches break collision domains to provide a faster and collision free network. In this chapter, we take a deeper look at how switches work.

From Chapter 4, you will remember that routing protocols are prone to loops. Similarly, redundant links in layer 2 can cause loops. By now you are aware that loops of any kind are bad. Hence, the Spanning Tree Protocol (STP) was developed to keep layer 2 networks loop free. STP is discussed in depth in this chapter.

- [6-1 Understanding Switching and Switches](#)
- [6-2 Initial Configuration of a Catalyst Switch](#)
- [6-3 Spanning Tree Protocol \(STP\)](#)
- [6-4 Cisco's additions to STP \(Portfast, BPDUGuard, BPDUFilter, UplinkFast, BackboneFast\)](#)
- [6-5 Rapid Spanning Tree Protocol \(RSTP\) – 802.1w](#)
- [6-6 Per-VLAN Spanning Tree Plus \(PVST+\) and Per-VLAN RSTP \(Rapid-PVST\)](#)
- [6-7 EtherChannel](#)
- [6-8 Lab 6-1 – Port Security](#)
- [6-9 Lab 6-2 – STP](#)

## **6-1 Understanding Switching and Switches**

### **History of Switching**

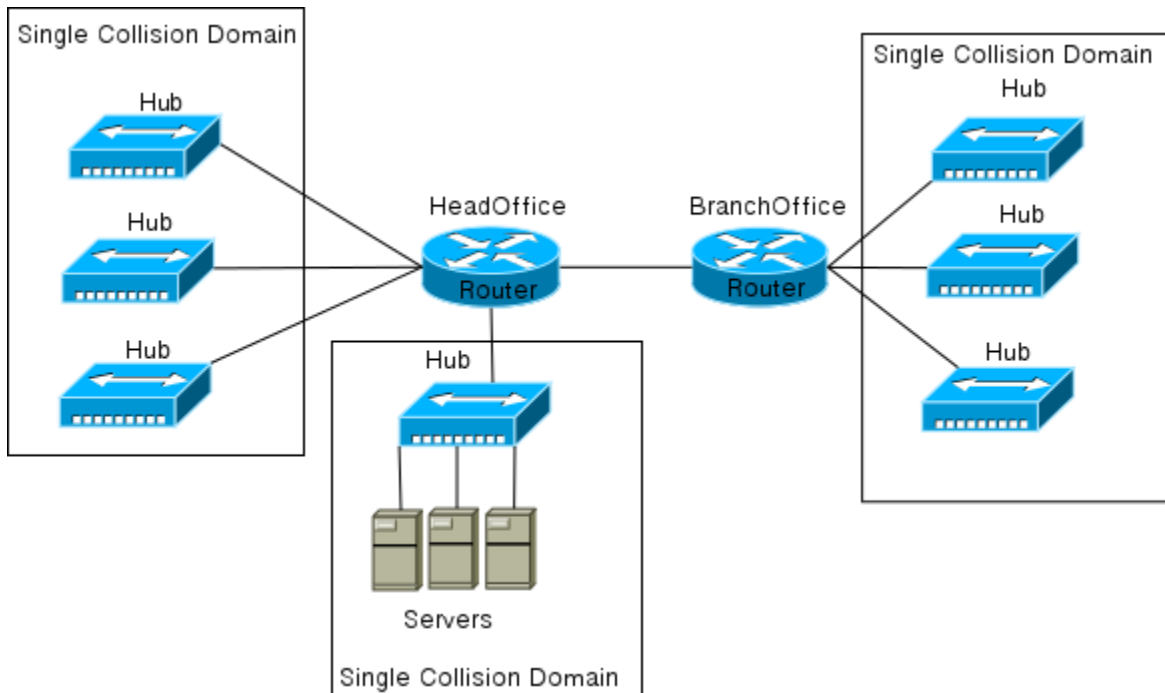
To understand the importance of current day switches, you need to understand how networks used to work before switches were invented. During mid to late 1980s 10Base2 Ethernet was the dominant 10Mbps Ethernet standard. This standard used thin coaxial cables with a maximum length of 185 meters with a maximum of 30 hosts connected to the cable. Hosts were connected to the cable using a T-connector. Most of the hosts at that time were either dumb terminals or early PCs that connected to a mainframe for accessing services.

When Novell became very popular in the late 80s and early 90s, NetWare servers replaced the then popular OS/2 and LAN Manager servers. This made Ethernet more popular, because Novell servers used it to communicate between clients and the server. Increasing dependence on Ethernet and the fact that 10Base2 technology was costly and slow lead to rapid development on Ethernet. Hubs were added to networks so that the 10Base5 standard could be used with one host and a Hub port connected on each cable. This led to collapsed backbone networks such as the one shown in Figure 6-1.

As you already know, networks made of only Hubs suffer from problems such as broadcast storms and become slow and sluggish. The networks of late 80s and early 90s suffered from the same problem. Meanwhile, the dependence on networks and services available grew rapidly. The corporate network became huge and very slow since most of the services were available on it and remote offices depended on these

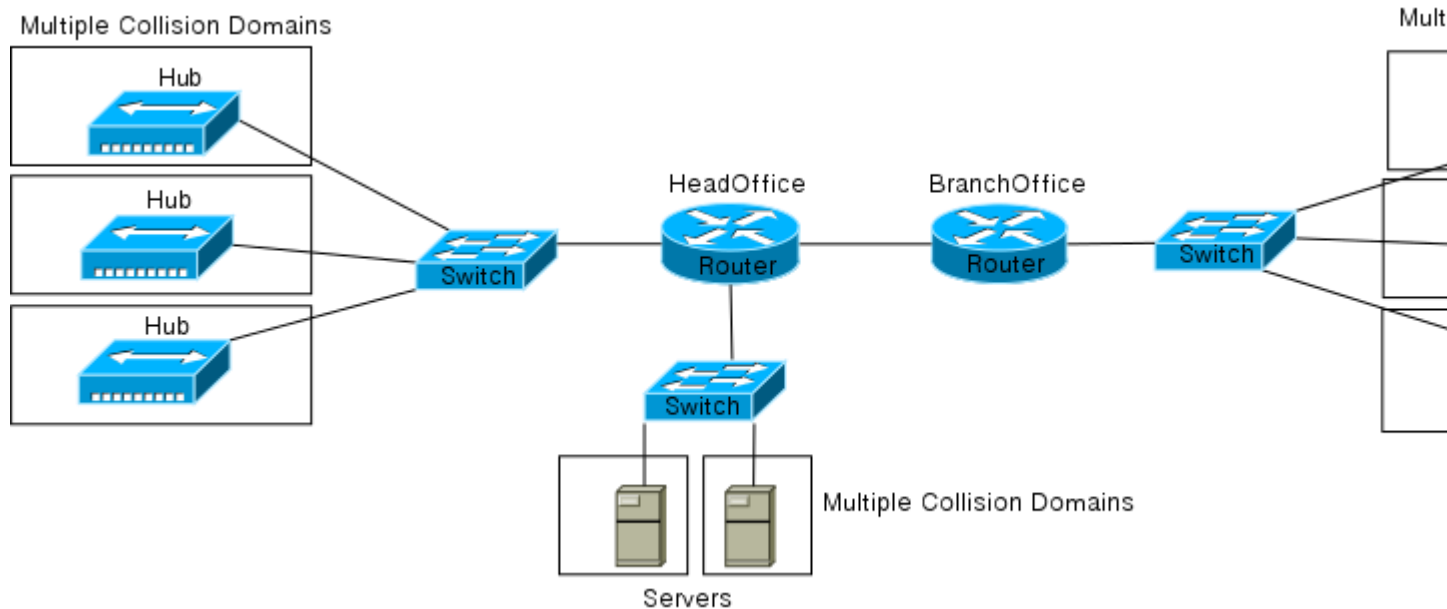
services. Segmenting the networks and increasing their bandwidth became a priority. With the introduction of devices called **bridges**, some segmentation was introduced. Bridges broke up collision domain but were limited by the number of ports available and the fact that they could not do much apart from breaking up the collision domains.

**Figure 6-1** Collapsed Backbone Network



To overcome the limitations of bridges, **switches** were invented. Switches were multiport bridges that broke the collision domains on each port and could provide many more services than bridges. The problem with the early switches was that they were very costly. This prohibited connecting each individual host to a switch port. So after introduction of switches, the networks came to look like the one shown in Figure 6-2.

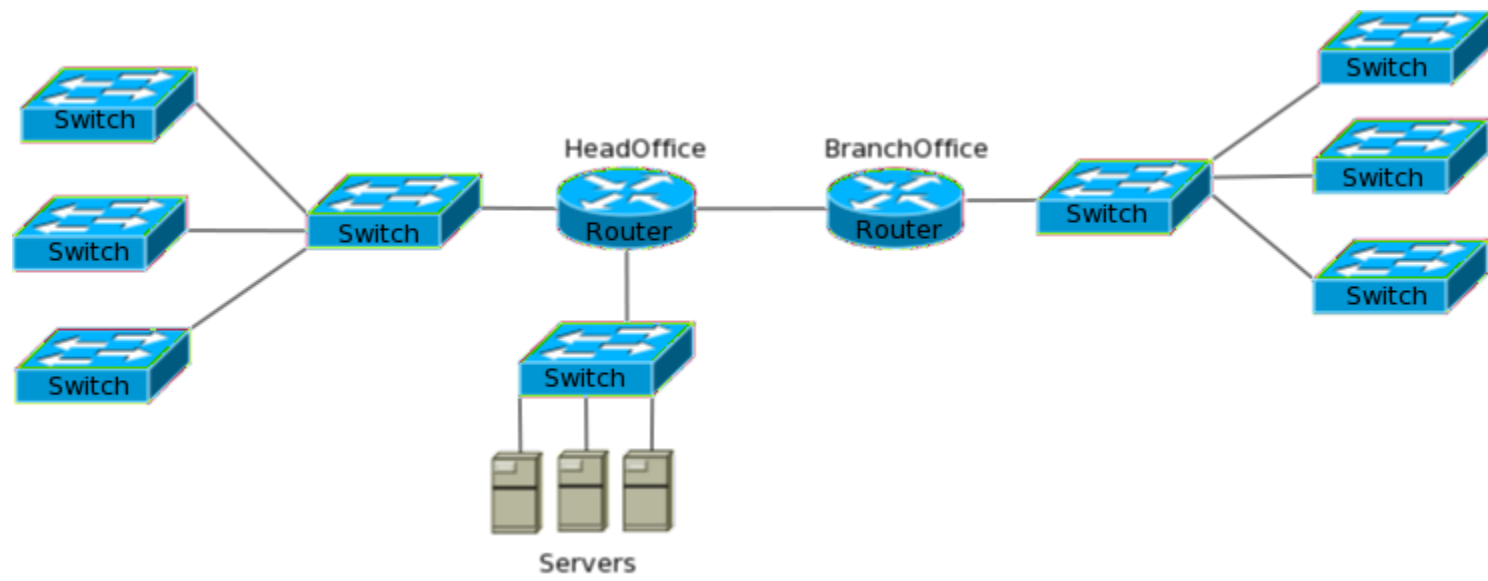
**Figure 6-2** Early Switched Networks



In these networks, each hub was connected to a Switch port. This change increased the network performance greatly since each hub now had its own collision domain instead of the entire local network being a single collision domain. Such networks, though vastly better than what existed earlier, still forced hosts connected to hubs to share a collision domain. This prevented the networks from attaining their potential. With the drop in prices of switches, this final barrier was also brought down. Cheaper switches meant that each host could finally be connected to a switch port thereby, providing a separate collision domain for each host. Networks came to look like the one shown in Figure 6-3. Such networks practically had no collision.

**Figure 6-3** Switched networks





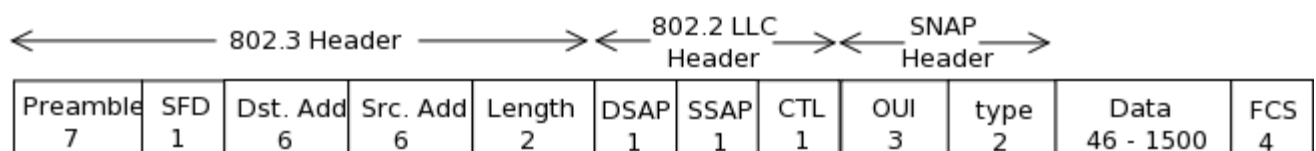
## Understanding Switches and their limitations

While switches are multiport bridges, these two devices have a significant difference. Bridges use software to build and maintain the switching and filtering tables while Switches use the hardware – more specifically **application specific integrated circuits (ASICs)** to build and maintain their tables. Both of these devices provide a dedicated collision domain on each of their ports but switches go a little further by providing the following features:

- **Lower latency** – Since switches used hardware based bridging using ASICs, they work faster than software based bridges.
- **Wire Speed** – Hardware based switching allows for a near wire speed functionality due to low processing time.
- **Low cost** – Cost of switches is very low, making cost connecting each host cost very low.

So what makes layer 2 switching this fast and efficient? The fact that layer 2 switching is hardware based and only looks at the hardware address in each frame before deciding on an action. Layer 3 routing on the other needs to look at Layer 3 header information before making a decision. You will remember from chapter 1 that the destination MAC address starts from the 9<sup>th</sup> byte of a packet and is 6 bytes long. So the switch only has to read 14 bytes when a frame is received. Figure 6-4 shows a frame.

**Figure 6-4** *Ethernet Frame*



Overall, using switches to segment networks and provide connectivity to hosts results in very fast and efficient network with each host getting the full bandwidth.

While switches increase the efficiency of the network, they still have the limitations discussed below:

1. While switches break collision domains, they do not break broadcast domains. The entire layer 2 network still remains a single broadcast domain. This makes the network susceptible to broadcast storms and related problems. Routers have to be used to break the broadcast domains.
2. When redundancy is introduced in the switched network, the possibility of loops becomes very high. Dedicated protocols need to be run to ensure that the network remains loop free. This increases burden on the switches. The convergence time of these protocols is also a concern since the network will not be useable during convergence.

Due to the above limitations, routers cannot be eliminated from the network. To design a good switched or bridged network, the following two important points must be considered:

1. Collision domain should be broken as much as possible.
2. The users should spend 80 percent of their time on the local segment.

### **Bridging vs switching**

While switches are just multiport bridges, there are many differences between them:

1. Bridges are software based while switches are hardware based since they use ASICs for building and maintaining their tables.
2. Switches have higher number of ports than bridges
3. Bridges have a single spanning tree instance while switches can be multiple instances. (Spanning tree will be covered later in the chapter).

While different in some aspects, switches and bridges share the following characteristics:

1. Both look at hardware address of the frame to make a decision.
2. Both learn MAC address from frames received.

3. Both forward layer two broadcasts.

### Three functions of a switch

A switch at layer 2 has the following three distinct functions:

1. Learning MAC addresses
2. Filtering and forwarding frames
3. Preventing loops on the network

It is important to understand and remember each of these three functions. The following sections explain these three functions in depth.

### Learning MAC Addresses

When a switch is first powered up it is not aware of the location of any host on the network. In a very short time, as hosts transmit data to other hosts, it learns the MAC address from the received frame and remembers which hosts are connected to which port.

If the switch receives a frame destined to an unknown address, it will send a broadcast message out of each port except the port that the request was received on, and then when the switch receives a reply it will add the address and source port to its database. When another frame destined to this address is received, the switch does not need to send a broadcast since it already knows where the destination address is located.

You can see how switches differ from hubs. A hub will never remember which hosts are connected to which ports and will always flood traffic out of each and every port.

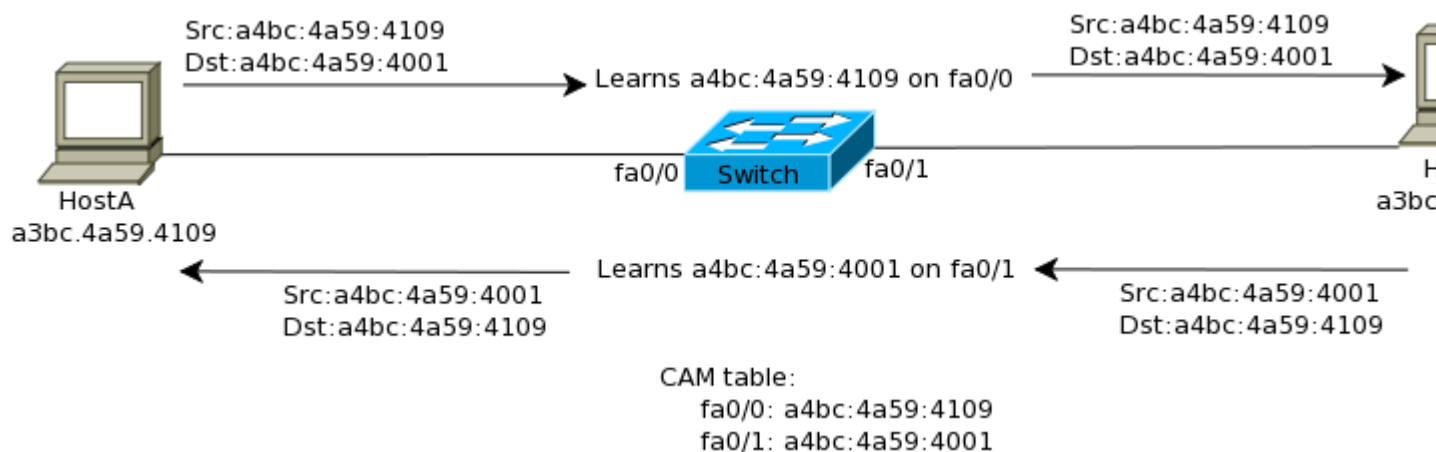
The table in which the addresses are stored is known as CAM (Content-addressable memory) table. To further understand how the switch populates the CAM table, consider the following example:

1. A switch boots up and has an empty CAM table.

2. HostA with a MAC address of a3bc.4a59.4109 sends a frame to HostB whose address is a3bc.4a59.4001.
3. The switch receives the frame on interface fa0/1 and saves the MAC address of HostA (a3bc.4a59.4109) in its CAM table and associates it with interface fa0/1.
4. Since the destination address is not known, the switch will broadcast the frame out all interfaces except fa0/1.
5. HostB receives the frame and replies back.
6. The switch receives the reply on interface fa0/2 and saves the MAC address of HostB (a3bc.4a59.4001) in its CAM table and associates it with interface fa0/2.
7. The switch forwards the frame out interface fa0/1 since the destination MAC address (a3bc.4a59.4109) is present in the CAM table and associated with interface fa0/1.
8. HostA replies back to HostB.
9. The switch receives the frame and forwards it out interface fa0/2 because it has the destination MAC address (a3bc.4a59.4001) associated with interface fa0/2 in the CAM table.

The above exchange is illustrated in Figure 6-5.

**Figure 6-5** Switch learning MAC addresses



The switch will store a MAC addresses in the CAM table for a limited amount of time. If no traffic is heard from that port for a predefined period of time then the entry is purged

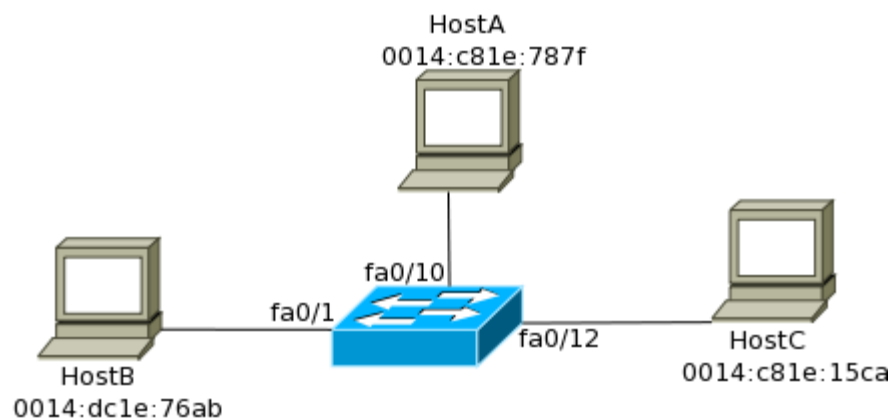
from memory. This is to free up memory space on the switch and also prevent entries from becoming out of date and inaccurate. This time is known as the MAC address aging time. On Cisco 2950 this time is 300 seconds by default and can be configured to be between 10 and 1000000 seconds. The switch can also be configured so as to not purge the addresses ever.

The command to see the CAM table of a Switch is “**show mac address-table**”. Here is an example of how the CAM table of a Switch:

The CAM table shown above will be created for the network shown in Figure 6-6.

Note that the switch stores the Mac Address and the Port where the host is connected. Do not worry about the VLAN column at the moment; we will cover this in chapter 7.

**Figure 6-6** *CAM table*



### Filtering and Forwarding frames

When a frame arrives at a switch port, the switch examines its database of MAC addresses. If the destination address is in the database the frame will only be sent out of the interface the destination host is attached to. This process is known as **frame filtering**. Frame filtering helps preserve the bandwidth since the frame is only sent out the interface on which the destination MAC address is connected. This also adds a layer of security since no other host will ever receive the frame.

On the other hand, if the switch does not know the destination MAC address, it will flood the frame out all active interfaces except the interface where the frame was received on. Another situation where the switch will flood out a frame is when a host sends a broadcast message. Remember that a switched network is a single broadcast domain.

Let's take two examples to understand frame filtering. A switch's CAM table is shown below:

When it receives a frame from fa0/1 destined for a host with MAC address of 0014.c8ef.19fa, what will the switch do? Since the address is not known, switch will flood out the frame out all active interfaces except fa0/1. If a response is received from the destination host, the MAC address will be added to the CAM table.

In another example, if a host with MAC address of 0014.c8ef.20ae, connected to interface fa0/11 sends a frame destined to 0014.bc1e.76ab, the switch will add the source address to its CAM table and associate it with fa0/11. It will then forward the frame out fa0/1 since the destination address exists in the CAM table and is associated with fa0/1.

The CAM table of the switch will now look like the following:

### Switching Methods

Any delay in passing traffic is known as latency. Cisco switches offer three ways to switch the traffic depending upon how thoroughly you want the frame to be checked before it is passed on. The more checking you want the more latency you will introduce to the switch.

The three switching modes to choose from are:

- Cut through
- Store-and-forward
- Fragment-free

### Cut-through

Cut-through switching is the fastest switching method meaning it has the lowest latency. The incoming frame is read up to the destination MAC address. Once it reaches the destination MAC address, the switch then checks its CAM table for the correct port to forward the frame out of and sends it on its way. There is no error checking so this method gives you the lowest latency. The price however is that the switch will forward any frames containing errors.



The process of switching modes can best be described by using a metaphor.

You are the security at a club and are asked to make sure that everyone who enters has a picture ID. You are not asked to make sure the picture matches the person, only that the ID has a picture. With this method of checking, people are surely going to move quickly to enter the establishment. This is how cut-through switching works.

### **Store-and-forward**

Here the switch reads the entire frame and copies it into its buffers. A cyclic redundancy check (CRC) takes place to check the frame for any errors. If errors are found the frame is dropped otherwise the switching table is examined and the frame forwarded. Store and Forward ensures that the frame is at least 64 bytes and no larger than 1518 bytes. If smaller than 64 bytes or larger than 1518 bytes then the switch will discard the frame.

Now imagine you are the security at the club, only this time you have to not only make sure that the picture matches the person, but you must also write down the name and address of everyone before they can enter. Doing it this way causes a great deal of time and delay and this is how the store-and-forward method of switching works.

Store-and-forward switching has the highest latency of all switching methods and is the default setting of the 2900 series switches.

### **Fragment-free (modified cut-through/runt-free)**

Since cut-through can ensure that all frames are good and store-and-forward takes too long, we need a method that is both quick and reliable. Using our example of the nightclub security, imagine you are asked to make sure that everyone has an ID and that the picture matches the person. With this method you have made sure everyone is who they say they are, but you do not have to take down all the information. In switching we accomplish this by using the fragment-free method of switching.

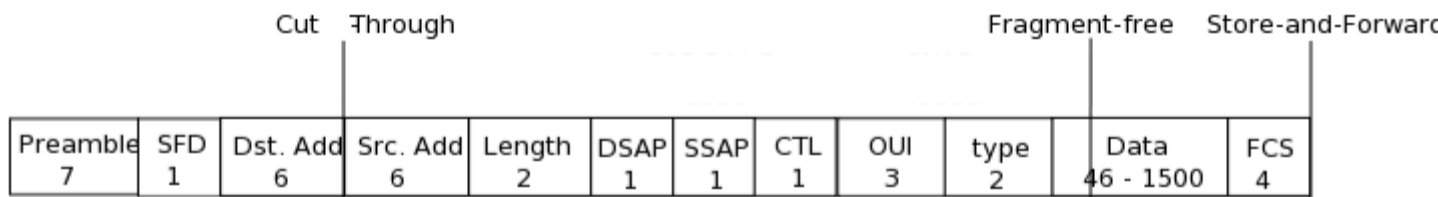
This is the default configuration on lower level Cisco switches. Fragment-free, or modified cut-through, is a modified variety of cut-through switching. The first 64 bytes of

a frame are examined for any errors, and if none are detected, it will pass it. The reason for this is that if there is an error in the frame it is most likely to be in the first 64 bytes.

The minimum size of an Ethernet frame is 64 bytes; anything less than 64 bytes is called a “runt” frame. Since every frame must be at least 64 bytes before forwarding, this will eliminate the runts, and that is why this method is also known as “runt-free” switching.

The figure below shows which method reads how much of a frame before forwarding it.

*Different Switching methods*

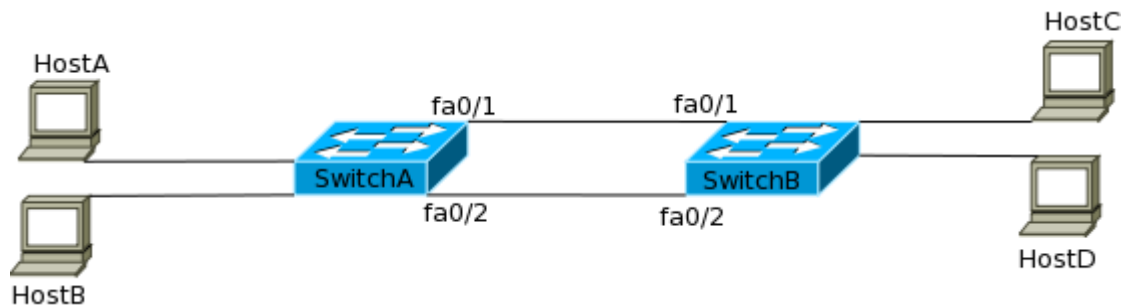


## Preventing loops in the network

Having redundant links between switches can be very useful. If one path breaks, the traffic can take an alternative path. Though redundant paths are extremely useful, they often cause a lot of problems. Some of the problems associated with such loops are broadcast storms, endless looping, duplicate frames and faulty CAM tables. Let’s take a look at each of these problems in detail:

- **Broadcast Storms** – Without loop avoidance techniques in place, switches can endlessly flood a broadcast in the network. To understand how this can happen, consider the network shown in Figure 6-7.

**Figure 6-7** *Broadcast Storms*



In the network shown in Figure 6-7, consider a situation where HostA send out a broadcast. The following sequence of events will then happen:

- SwitchA will forward the frame out all interface except the one connected to HostA. HostB will receive a copy of this broadcast. Notice that the frame would have gone out of interface fa0/1 and fa0/2 also. For ease of understanding lets call the frame going out of fa0/1 as frame1 while the frame going out of fa0/2 frame2.
- When SwitchB receives frame1, it will flood it out all interfaces including fa0/2. When it receives frame2, it will flood it out all interfaces including fa0/1. HostC and HostD would receive both the frames, which actually means they receive two copies of the same frame. Meanwhile one frame was each sent out fa0/2 and fa0/1 towards SwitchA! Let's call these frames frame3 and frame4.
- When SwitchA receives frame3, it will flood it out all interfaces including fa0/1 and when it receives frame4 it will flood it out all interface including fa0/2. This means, HostB and HostA both receive two broadcasts. Remember that HostA was the original source of the broadcast while HostB has already receive one copy! But the worse part is that two more frames went out to SwitchB. Now the previous and the current step will continue endlessly and the four hosts will be continuously get the broadcast.

If multiple broadcasts are sent out to this network, each of them will endlessly be sent to every host in the network thereby causing what is known as a broadcast storm.

- **Endless looping** – Similar to what happens in a broadcast storm, consider a situation where HostA in Figure 6-7 sends a unicast destined to a host which does not exist in the network. SwitchA will receive the frame and will see that it

does not know the destination address. It will forward it out all interface except the one where HostA is connected. SwitchB will receive two copies of this frame and will flood them out all its interfaces since it does not know the destination address. Since SwitchB will flood the frames out fa0/1 and fa0/2, SwitchA will receive the frames and the endless loop will continue.

- **Duplicate frames** – In the network shown in Figure 6-7, consider a situation where HostA sends a frame destined to HostD. When switch A receives this frame, it will not know where HostD is and will flood it out all the interfaces. SwitchB will receive one copy each from both fa0/1 and fa0/2 interfaces. It will check the destination address and send both the packets to HostD. In effect, HostD would have received a duplicate packet. This might cause problems with protocols using UDP and especially with voice packets.
- **Faulty CAM table** – Consider the situation where HostA sends a frame destined to HostC. When SwitchA receives the frame, it does not know where HostC resides, so it will flood out the frame. SwitchB will receive the frame on both fa0/1 and fa0/2. It will read the source address and store it in its CAM table. Now it has two destination interfaces for a single address! Now a switch cannot have two entries for a single address, so it will keep overwriting each entry with new information as frames are received on multiple interfaces. This can cause the switch to get overwhelmed and it might stop forwarding traffic.

All of these problems can cause a switched network to come crashing down. They should be entirely avoided or at least fixed. Hence, the Spanning Tree Protocol was created to keep the network loop free. We will be discussing STP shortly.

## **6-2 Initial Configuration of a Catalyst Switch**

The process to connect to the CLI of a catalyst switch and the initial configuration was covered in detail in Chapter 3. I would recommend reading that chapter again to get familiar with the CLI of a switch. The list below briefly covers some initial configuration steps to get you started.

- **Hostname** – You can set the name of the device with the **hostname** command in the global configuration mode. Setting the name of the device does not have any impact on the functions of the switch. It will continue to perform normally irrespective of the name but it is easier to manage and troubleshoot your network when you give the devices a

meaningful name. The example below shows how you can change the hostname. Notice the immediate change in prompt after the command is executed.

- **Clock** – You can set the date and time on the switch with the **clock** command in the privileged exec mode. Setting the correct date and time is a requirement for some advanced configuration but it helps when troubleshooting the device. The syntax of the command is **clock set** *hh:mm:ss day month*. An example is shown below:
- **Setting enable secret** – The enable secret allows setting a password for access to the privileged mode. As you know the privileged mode is where most configuration changes can be made. It can be set using the **enable secret** command in global configuration mode as shown below:
- **Securing access to CLI** – As you already know, the switch CLI can be accessed using the console, vty or aux lines. These can be secure by setting a password so that only authorized users can connect. The password can be set using the **password** command in the line mode as shown below:

One thing you must remember is that the interface configuration on a switch differs greatly from the interface configuration of a router because switch interfaces are layer 2 interfaces (called switchports) unlike router interfaces which are layer 3 interfaces. Chapter 6 and Chapter 7 cover various interface level configuration for the Switch. The command to enter the interface configuration mode remains the same on the router as shown below:

## Port Security

Typically, the Switch will learn the MAC address of the device directly connected to a particular port and allow traffic through. This behavior can be a huge security risk if an intruder manages to connect a host to your switchport. At some stage (and in CCNA!) you will need to restrict who can connect to the switched network. This is where port security can assist us. Cisco switches allow us to control which devices can connect to a switch port or how many of them can connect to it (such as when a hub or another switch is connected to the port).

Port security is disabled by default. Before configuring the Port Security, we have to enable it. It can be enabled using the **switchport port-security** command. Here's how to do it:

As soon as port security is enabled, it will apply the default values, which is one host permitted to connect at a time. If this rule is violated the port will shutdown.

Using the port security feature we can specify:

1. Who can connect to the Switchport
2. How many can connect to the Switchport
3. Violation Action

Let's take a look at all the three options:

**Who can connect** – If you know that only a particular host should be connecting to a switchport, then you can restrict access on that port to the MAC address of that host. This will ensure that no one can unplug the authorized host and connect another one. This is a good option for secure locations. This is done using the following command:

Example: If we want only the host with MAC address 0001.14ac.3298 to connect to port fa0/10 on our switch, then the commands required will be:

You have to remember that this command will not add the MAC address to the CAM table. When a host connects to this port and sends the first frame, the source address of the frame is checked against the configured MAC address. If a match is found that the address is added to the CAM table.

So do we have to provide each host's MAC address manually? That's a huge task considering thousands of hosts that a network can have! Well, not really. Port security provides something called a sticky address. The Switch will use the MAC-address of the



first host connected to the port as a static MAC-address and only that host will be able to connect to the port subsequently. The command required is:

**How many can connect** – Let's say we have only one switchport left free and we need to connect 5 hosts to it. What can we do? Connect a Hub or Switch to the free port! Connecting a switch or a hub to a port has implications. It means that the network will have more traffic. If a user instead of an administrator connects a switch or a hub then there are chances that loops will be created. So it is best that number of hosts allowed to connect is restricted at the switch level. This can be done using the “**switchport port-security maximum**” command. This command configures the maximum number of MAC addresses that can source traffic through a port. Consider the following examples:

- **Example 1** – Allow only one host to connect to the port. Learn the MAC address of the allowed host automatically.
- **Example 2** – Allow 3 hosts to connect at the same time out of which 1 MAC address is static and the other two can vary.
- **Example 3** – Allow a maximum of 5 hosts to connect simultaneously. Hosts can vary.

**Violation Action** – What happens if a violation of security occurs on a switchport? What if 5 hosts are allowed on a port but 6 connect to it? The switch can take one of the three configured actions:

- Shutdown the port.
- Keep the port up but do not allow the offending host to send/receive data (protect).
- Keep the port up but do not allow the offending host to send/receive data and notify the administrator through SNMP and/or syslog. (restrict).

The three modes can be configured using the following commands:

Let's verify our port security configuration using the “**show port-security interface**” command:

The above out shows that Fa0/1 has been configured with 3 static MAC Addresses and will allow a maximum of 5 hosts to connect to it. If a violation is detected then the port (by default) will go into error-disabled mode and shut the port (switch interface) down.

You can see this happening on the below switch where an unauthorized MAC address comes into the fast Ethernet 0/2port.

Another important command is “**show port-security**” command. This command provides an overview of all the ports that have port security configured:

## **6-3 Spanning Tree Protocol (STP)**

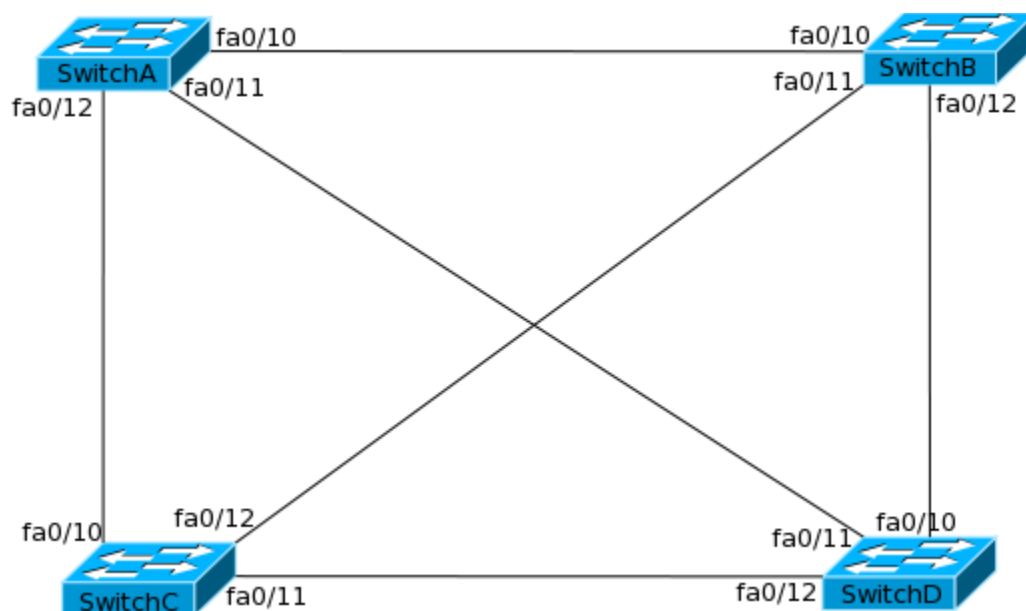
Figure 6-8 shows a full mesh network. A good redundant setup where, if one link fails, there would be two more links for traffic to go through. However, could this lead to any problems? Let's say a host is connected to port fa0/1 on SwitchA (not shown) and this switch sends a broadcast out to the network. SwitchA has to forward this frame out every port except fa0/1. A part of what happens next is shown below:

1. SwitchB receives the packet on fa0/10 and sends it out on every port except that one.

2. SwitchD receives the packet on fa0/10 and sends it out on every port except that one but including fa0/11.
3. SwitchA receives the packet on fa0/11 and sends it out on every port except fa0/11 but including fa0/1 and fa0/10!

What we see here is that not only the original source received the frame back but now SwitchA has to send the packet back out fa0/10 also. Back to the step one to three which goes on forever.

**Figure 6-8** Full mesh switched network



As you already know, what we have just seen is a loop and such loops can bring a network to a grinding halt. Layer 2 LAN protocols have no method to stop traffic endlessly travelling around possibly carrying inaccurate information. At layer 3 we can make packets expire after a certain amount of time or after they have traveled a certain distance (using route poisoning for example – see the routing module for more info).

As layer 2 networks grew, it quickly became evident that a system to prevent loops was needed if LANs were to continue to function. Digital Equipment Corporation created a protocol called **Spanning Tree Protocol (STP)** to prevent broadcast storms and network loops at layer 2. The IEEE under standard 802.1d now regulates STP.

STP allows bridges and switches to communicate with each other so they can create a loop free topology. Each bridge runs the Spanning Tree Algorithm that calculates how a loop can be prevented. When STP is applied to a looped LAN topology, all segments will be reachable but any open ports that would create a traffic loop are blocked. When it sees a loop in the network it blocks one or more redundant paths preventing a loop from forming. STP continually monitors the network always looking for failures on switch ports or changes in the network topology. If a change on the LAN is detected, STP can quickly make redundant ports available and close other ports to ensure the network continues to function normally.

Before we learn further about STP, we need to understand some of the common terms associated with it.

**Bridge ID:** This is a unique identification number of each switch in the network. It consists of bridge priority and the base MAC Address of the switch. The default bridge priority of a Cisco Switch is 32768. This is a configurable value between 0 to 61440 but the value has to be in increments of 4096. 4096, 8192, 12288, so on and so forth are acceptable values. Priority plays a very big role in STP and how well the network will function.

**Root Bridge:** All switches in the network elect the root of the tree. Thereon all decisions such as which redundant path to block and which to open are taken from the perspective of the root switch (commonly called the Root Bridge). The switch with the lowest Bridge ID wins the election. Switches that do not become Root Bridge are called NonRoot Bridges.

**BPDU:** Bridge Protocol Data Unit (BPDU) is the information exchanged between switches to select the Root Bridge as well as configure the network after that. A decision on which port to block is taken after examining BPDUs from the neighbors. Cisco Switches send BPDUs every 2 seconds by default. This value can be configured from 1 second to 10 seconds.

**Root Port:** Each switch has to have a path to the Root Bridge, if not directly connected. Root port is the directly connected link or the fastest path to the Root Bridge from a NonRoot bridge.

**Port Cost:** Each port has a cost that is determined by the bandwidth of the link. Port cost determines which of the redundant links will not be blocked. The lower the cost, the better it is. Port Cost also determines which port will become the root port if multiple paths to the root bridge exist. Default port costs are shown below.

**Table 6-1** *Default STP cost*

Link Speed	STP cost
4 Mbps	250
10 Mbps	100
16 Mbps	62
100 Mbps	19
1 Gbps	4
10 Gbps	2

**Designated Port:** The bridges on a network segment collectively determine which bridge has the least-cost path from the network segment to the root. The port connecting this bridge to the network segment is then the designated port for the segment. Ports that are not selected Designated Ports are called Non-Designated Ports.

### Port States in Spanning Tree

Switch ports running STP can be in one of five states.

- Blocked
- Listening
- Learning
- Forwarding
- Disabled

STP port states are very important. You should remember these states and what they mean. Each of them is discussed below.

### **Blocked**

None of the ports will transmit or receive any data, but they will listen to BPDUs. The BPDU carries various pieces of information that are used by STP to determine what state the ports should be in and what the STP topology should be.

### **Listening**

The switch listens for frames but doesn't learn or act on them. The switch does receive the frames but discards them before any action is taken. MAC addresses are not placed into the CAM table while the port is listening.

### **Learning**

The switch will start to learn MAC addresses it can see and will populate its CAM table with the addresses and the ports on which they were found. In this state, the switch will start to transmit its own BPDUs.

### **Forwarding**

The switch has learned MAC addresses and corresponding ports and populates its CAM table with this. The switch can now forward traffic.

### **Disabled**

In the Disabled state, the port will receive BPDUs but will not forward them to the switch processor. It discards all incoming frames from both the port and other forwarding ports on the switch.

The port states are transitional and allow other BPDUs to arrive in good time from other switches. Port transition times are typically:



- Initialization to blocking
- Blocking to listening (20 secs)
- Listening to learning (15 secs)
- Learning to forwarding (15 secs)
- Forwarding to disabled (if there is a failure)

All ports start at the blocking state (there are a few exceptions discussed later). After STP convergence, some ports will transition to listening, learning, and finally forwarding while the rest would remain in a blocked state. Thus the time needed to transition from one stage to another; we find that a layer 2 network running STP takes 50 seconds to start switching data! This is known as the convergence time.

### STP Convergence

Remember that Spanning tree works by selecting a root bridge on the LAN. It is selected by comparing Bridge ID of each switch.

STP is considered to be converged after three steps have taken place:

- Elect root bridge
- Elect root ports
- Elect designated ports

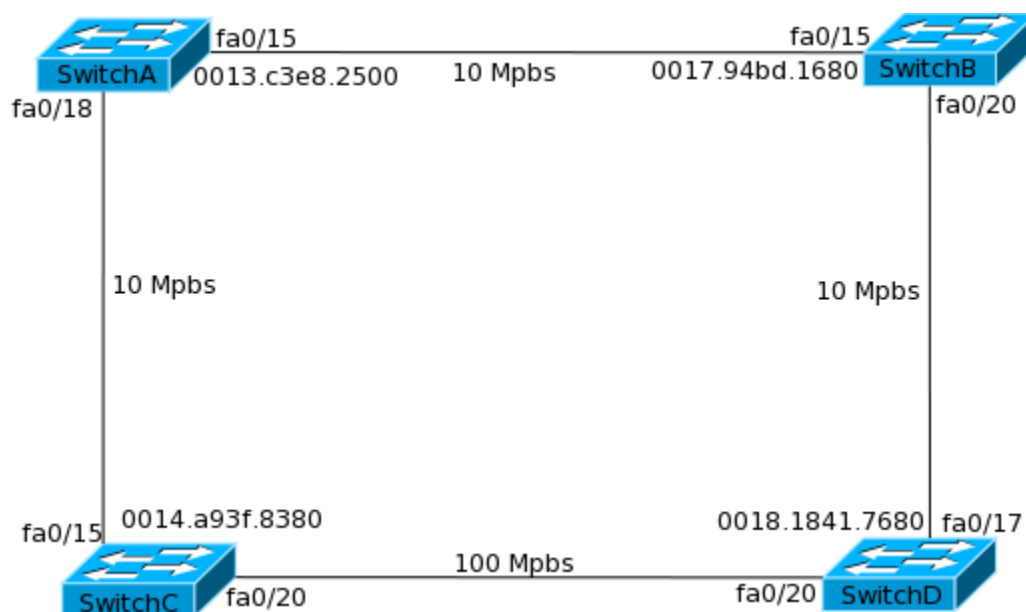
Each of the above three steps are discussed in detail below. The network shown in figure 6-9 will be used to explain the STP convergence process.

### Elect Root Bridge

The bridge with the lowest Bridge ID (BID) becomes the root bridge. The BID consists of two values in an 8-byte field. The bridge priority (32,768 by default) makes up two bytes and the MAC address of the backplane or supervisor module (depending upon the model of switch) makes up the rest of the six bytes.

The root bridge on a LAN is selected by an election. Each switch running STP passes information in a format known as bridge protocol data units (BPDUs). BPDUs are multicast frames that can be thought of as hello messages between STP enabled switches and they are sent out every two seconds from every port. This is necessary to maintain a loop free topology. When the switch or bridge priorities combined with its MAC address are all exchanged; the bridge with the lowest ID is selected as the root bridge.

**Figure 6-9** STP Convergence



All ports on the root bridge are set as designated and thus are always set to a forwarding state.

In our network, the priority of all the switches has been left at the default value. So the switch with the lowest MAC address will be selected the root bridge. In this case it will be SwitchA.

To verify this we issue the “**show spanning tree vlan (vlan#)**” command on SwitchA :

In the above output notice that the fourth line states that this bridge is the root bridge. At this stage do not worry about the number 5 used in the command. That is the VLAN id and will be discussed in chapter 7.

Now if we want SwitchC to be the root bridge then we will need to give it better priority using the following command:

Let's check the "show spanning-tree" output now on SwitchC and SwitchA

In the above output notice that:

- SwitchC shows it is the root bridge now.
- 1. SwitchA shows the MAC address of SwitchC as that of the root bridge along with the new priority of SwitchC.

In case you are wondering why SwitchC's priority is 8197 instead of 8192, we will come to this point shortly. Let's set the priority on SwitchC back to 32768 and make SwitchA the root bridge for the following sections.

### **Elect Root Ports**

For non-root bridges there will be only one root port. The root port will be the port with the lowest path cost to the root bridge. The root port will also be set to forwarding state.

Path cost is the cost of transmitting a frame to the root bridge. The value is set according to the bandwidth of the link on the LAN. The slower the link, the higher the cost is.

In our network, SwitchB and SwitchC's fa0/15 ports will be the root ports because they are directly connected to SwitchA.

Switch D has two options – fa0/17 towards SwitchB and fa0/20 towards SwitchC. The total cost of the link on fa0/17 is 200 ( $2 \times 10 \text{ Mbps} = 100 \times 2$ ). The total cost of the link on fa0/20 is 119 ( $10 \text{ Mbps} = 100$  and  $100 \text{ Mbps} = 19$ ). So fa0/20 will be the root port for SwitchD and fa0/17 will be blocked. Remember that a default cost is associated with the bandwidth of a link. The default cost can be seen in table 6-1.

Let us verify SwitchD's root port using the "show spanning-tree" command:

Notice that the role for interface fa0/20 is shown as Root while the status is forwarding. On the other hand, fa0/17 is in the blocked state.

If we want to make fa0/17 on SwitchD as a root port instead of fa0/20, then we will need to change the cost on fa0/17 to something better (less) than 119. To do this, the “**spanning-tree cost**” command can be used on fa0/17. Look at the following output

In the above output notice that on changing the cost of interface fa0/17, it has become the Root port and is transitioning from Blocked to forwarding state while fa0/20 is now in the blocked state.

### **Elect Designated Ports**

If a switch has redundant ports connecting it to a LAN segment (another downstream switch or hub for example) then the port with the lowest cost will be elected the designated port. Designated ports forward BPDUs into the LAN segment and traffic to and from the LAN segment. In simple terms the designated port becomes the only link for the LAN segment towards the rest of the network and the root bridge.

In our example fa0/20 port on SwitchC will be the designated port for the link to SwitchD. If there were multiple links then an election would have taken place. Let us verify this on SwitchC:

Notice that fa0/20 has a role of designated port with state as forwarding. The election of designated port can be influence by changing the cost of the port This concludes a

basic overview of STP. STP can be difficult to understand and the following sections look deeper into various aspects of it. Hence, I strongly suggest you take a break and re-read this section to get a firm grasp of STP before continuing.

#### **6-4 Cisco's additions to STP (Portfast, BPDUGuard, BPDUFILTER, UplinkFast, BackboneFast)**



STP as we know it, keeps the network loop free but at what cost? The exact cost to you and I is 50 seconds! That is a long, long time in networking terms. For almost a minute data cannot flow across the network. In most cases this is a critical issue, especially for important network services.

To deal with this issue (before the industry standard was ratified) Cisco added the following features to STP implementation on its switches:

- PortFast, BPDUGuard and BPDUFilter
- UplinkFast
- BackboneFast

### Portfast

If you have a laptop or a server connected to a switchport then you know that:

- It will not need to listen to BPDUs because it is not a layer 2 device
- It will not create loops because it has a single link to the layer 2 network

Therefore, you can safely disable Spanning Tree on such ports. It is very important to ensure that such ports never have a STP enabled layer 2 device connected on them (Think port security!) or else a loop or a breakdown of the network is quite possible. You will even get a warning message on certain switches stating this when you enable portfast on a switchport!

When you configure a switchport as portfast, STP will be disabled on that port and it will transition to forwarding state when it comes up and will never be blocked.

The command to configure portfast is **spanning-tree portfast**:

As we learned, Portfast disables STP on a switchport but an important fact is that a Portfast switchport will keep listening for BDPUs. If someone adds a switch to a port which has been configured as Portfast, the consequences will be unpredictable and in some cases disastrous.

To guard against this situation, Cisco provides the BPDUGuard and BPDUFilter features.

### **BPDUGuard**

If a switch is plugged into a switchport configured as Portfast, it could change the STP topology without the administrator knowing and could even bring down the network. To prevent this, BPDUGuard can be configured on the switchport. With this configured, if a BPDU is received on a switchport, it will be put into an error disabled mode and an administrator will have to bring the port up. This can be configured on the port using the **“spanning-tree bpduguard enable”** command.

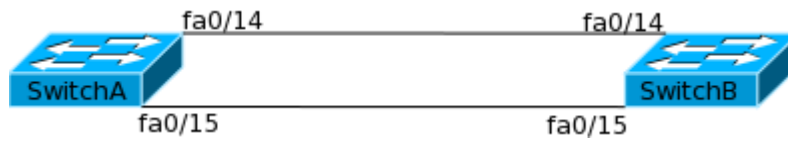
### **BPDUFilter**

When BPDUFilter is configured on a switchport which has been configured as Portfast, it will cause the port to lose the Portfast status if a BPDU is received on it. This will force the port to participate in STP convergence. This is unlike the behavior seen with BPDUGuard where the port is put into an error disabled mode. BPDUFilter can be enabled on the switchport using the **“spanning-tree bpdupfilter enable”** command.

### **UplinkFast**

To understand how UplinkFast helps speed up the convergence, consider the network shown in Figure 6-10. SwitchA is the Root Bridge in the network.

**Figure 6-10** *UplinkFast*



Now consider the following output from SwitchB

We will use the following debug commands on the switch.

These debugs will show us STP events and uplink fast messages. Now let's shut down port fa0/14 on SwitchB which is currently the root port as per output given above.

Note the time taken for fa0/15 to transition to forwarding state is 30 seconds. This is faster than the expected 50 seconds because listening and learning time were short in this P2P link between switches and no other hosts/switches are connected here.

Let's enable UplinkFast on SwitchB and repeat the process:

Note the time taken for fa0/15 to transition to forwarding is less than a second! From 30 seconds downtime to less than a second with UplinkFast enabled. Now that you have seen the difference it makes, let us define what exactly it does.

If a switch has multiple links towards the root bridge, then UplinkFast marks the redundant link as an Alternate Port and brings it up quickly in case the Root Port fails. This is possible because blocked ports keep listening for BDPUs.

Cisco recommends caution when using UplinkFast. You should enable it only on switches that have blocked ports.

### BackboneFast

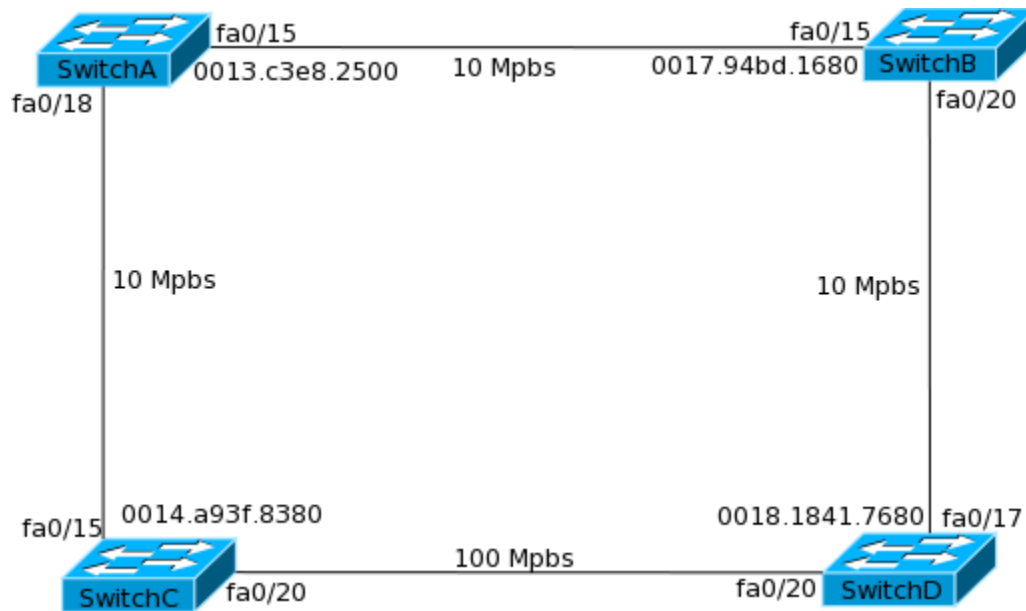
UplinkFast works by finding alternate ports for directly connected links. Similarly BackboneFast works on finding an alternate path when an indirect link to the root port goes down. To understand how BackboneFast works, consider the network shown in Figure 6-11. SwitchA is the Root Bridge here and Fa0/20 on SwitchD in the root port.

If SwitchC loses connection to SwitchA, it will advertise itself as the root bridge to SwitchD. SwitchD will compare previous known information with the new information and will learn that SwitchC has lost connection with SwitchA. Since the new BPDU states that a designated switch (SwitchC) is now the root bridge, this BPDU is known as inferior BPDU.

Eventually SwitchD will receive a BPDU from SwitchB stating the SwitchA is still the Root Bridge and SwitchD will now mark fa0/17 as the root port instead of fa0/20. This is because the information from SwitchB matches the existing information on SwitchD. BackboneFast ensures a quick failover as soon as the inferior BPDU is received. It saves roughly 20 seconds out of the 50 seconds of convergence time.

The **spanning-tree backbonefast** command can be used in the global configuration mode to enable BackboneFast as shown below:

**Figure 6-11** BackboneFast



### 6-5 Rapid Spanning Tree Protocol (RSTP) – 802.1w

The features discussed in the previous section – PortFast, UplinkFast and BackboneFast were added by Cisco and because of this they worked only on Cisco switches. IEEE added these features in a new STP protocol called Rapid Spanning Tree Protocol (RSTP) under the 802.1w standard.

One big different between 802.1D STP and 802.1w RSTP is that there are lesser number of port states. As you know, there are five states in 802.1D. RSTP only has 3 states. The disabled, blocking and learning states have been combined into a new discarding state in RSTP. Table 6-2 shows a comparison of the port states.

**Table 6-2** *STP and RSTP port states comparison*

<b>802.1D Port states</b>	<b>802.1w Port states</b>	<b>Is port active?</b>	<b>MAC addresses learned?</b>
<b>Disabled</b>	Discarding	No	No
<b>Blocking</b>	Discarding	No	No
<b>Listening</b>	Discarding	Yes	No
<b>Learning</b>	Learning	Yes	Yes
<b>Forwarding</b>	Forwarding	Yes	Yes

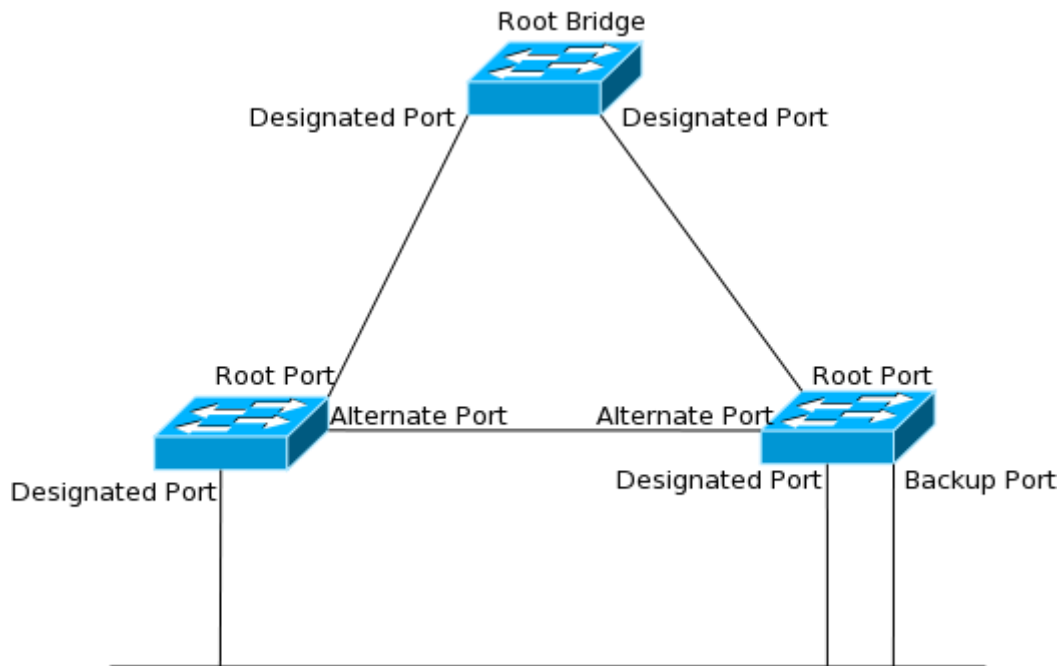
Similar to “traditional” spanning tree, RSTP will also elect a root bridge using the same parameters as STP and ports will be elected as root and designated ports. In addition to the standard root and designated ports, RSTP ports can have one of the following roles:

- **Alternate Port** – This is a port that provides an alternative path to the root bridge. This path is less desirable than the path provided by the root port but will be used if the path from the root port goes down.
- **Backup Port** – This is a port that provides a redundant path to a network segment but this path is less desirable than the one provided by the designated port. This path will be used if the path provided by the designated port goes down.

Figure 6-12 shows an example of a network with all port roles in RSTP.

**Figure 6-12** *RSTP Port Roles*





RSTP is backward compatible with 802.1D STP. If a switch with STP is discovered, the new features such as UplinkFast and BackboneFast will not be used.

Changing from 802.1D to 802.1w RSTP requires a single command on the switch – **spanning-tree mode rapid-pvst**. This is a global configuration mode command and will cause the switch to change to RSTP. Remember that this can cause the network to be temporarily unavailable. An example is shown below:

The change can be verified with the **show spanning-tree** command as shown below:

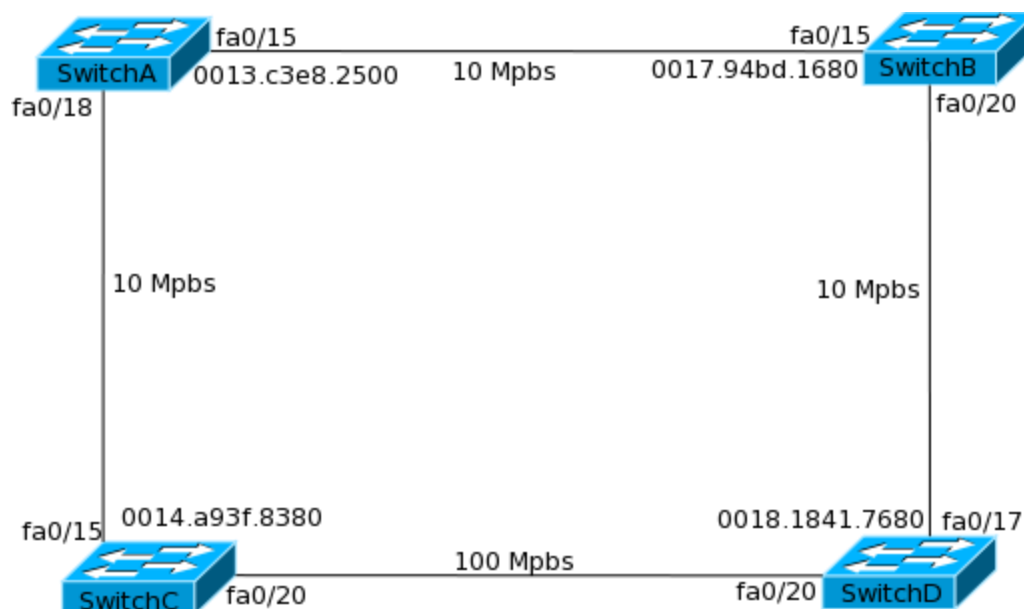


## **6-6 Per-VLAN Spanning Tree Plus (PVST+) and Per-VLAN RSTP (Rapid-PVST)**

This will be a good time to introduce you to another very significant change that Cisco made to STP. Before that, note that this section discusses VLANs. For now remember that VLANs provide different broadcast domains at layer 2 and hence keep traffic from one subnet different from another. We will cover VLANs in more detail in Chapter 7.

When IEEE 802.1d was drafted, VLANs did not exist. Hence one Spanning Tree instance worked across the entire switch. Eventually VLANs were introduced and they created different networks on the same switch. This gave rise to need to have different topology for load balancing and flexible Spanning Trees. The need for Per VLAN STP can be further understood from the network shown in Figure 6-13.

**Figure 6-13** *Per-VLAN STP*



SwitchD has two ways to paths to reach SwitchA. In any implementation of STP (Per-VLAN or a single STP), one of the interfaces will be blocked. Let us assume that fa0/17

is blocked in this network. This works well in an environment where the whole network is one single big network. Now consider a situation where the network is divided into two smaller networks using VLANs. If both the VLANs spanned all the four switches, would it not be useful to have fa0/17 blocked for one VLAN and fa0/20 blocked for the other VLAN? This way traffic in both VLANs can be load balanced across both paths!

To achieve this, Cisco added the Per-VLAN Spanning Tree Plus (PVST+) feature on its switches. With this feature, Cisco switches ran one STP instance for every VLAN.

When IEEE introduced 802.1w it still did not accommodate multiple Spanning Tree instances on a switch. Cisco introduced the Per-VLAN Rapid Spanning Tree (PVRST) to support Rapid Spanning Tree instances on each VLAN on the switch. PVST+ and PVRST both provide the same functionality across both 802.1D and 802.1w standards.

Remember that PVST+ and PVRST both add the VLAN number to the bridge ID of every switch. That is the reason you earlier saw the priority as 8197 in VLAN 5 even though you had configured the priority as 8192.

To enable RSTP for each VLAN in our switched network, we use the following command:

This is all that is needed if we need only instance of the spanning tree protocol. Later on in this section, we will show what is needed to enable the load sharing capabilities.

Using the “**show spanning-tree vlan <vlan#>**” command, we can verify which type of spanning tree is running.

Two items are of interest in this output. First is the Spanning Tree Protocol – RSTP and the second is the “sys-id-ext 10”. This shows that the Bridge priority was configured as 49152 and VLAN id 10 was added to it.

How can load balancing be achieved in the network shown in Figure 6-13 if VLAN 1 and VLAN 5 are being used on the LAN? We can achieve it by configuring Switch A with a better priority for VLAN 1 and configuring SwitchB with a better priority for VLAN 5. This can be done using the following commands:

Let's see the “show spanning-tree” output for both VLANs on SwitchD to verify loadbalancing.

We can see that the root bridge for VLAN1 is SwitchA whereas the root bridge for VLAN5 is SwitchB. Fa0/20 is the Root port for VLAN 1 and Fa0/17 is the root port for VLAN 5.

## **6-7 EtherChannel**

As you learned in the previous sections, STP essentially works by blocking redundant links between switches or segments. You may wonder why redundant links exist between switches or segments. They exist to provide a backup link in case a link fails. Apart from redundancy, these links can be used to provide extra bandwidth between these switches or segments. Instead of having a single link to transfer the data, traffic can be load balanced between two or more links. Since STP blocks all redundant links, the load balancing capability is lost.

To overcome this limitation of STP, EtherChannel is used to bundle up to 8 links into a single logical link. After the links are bundle, STP only sees a single logical link and is not able to block anything. Etherchannel protocols on the other hand provide methods to eliminate any loops within the physical links while load balancing traffic across them.

Etherchannel protocols also keep track of the status of each physical link. If one of the physical links go down or come back up, the protocols manage the deletion and addition of the link without STP realizing the change. Irrespective of status of physical links, STP only sees a single logical link.

Cisco switches can use the IEEE standard **Link Aggregation Control Protocol (LACP)** or Cisco's proprietary **Port Aggregation Protocol (PAGP)**. You have to know that each EthernChannel is called a channel group and a physical port can be added to it by using the **channel-group group-number mode on** command in the interface configuration mode. An example of this is shown below:

In the above example, Interfaces fa0/11 and fa0/12 on SW1 and SW2 are combined into an Etherchannel. Now SW1 and SW2 are connected via a single channel group consisting of two physical interfaces. Channel groups logical interfaces are presented as Po<group-number>. For example, the channel group created in the above example will be presented as interface Po1 since channel group number 1 was used. To verify the configuration you can see the output of **show interface trunk** as shown below:

To see if the effect of Etherchannel on STP, see the output of **show spanning-tree vlan** command as shown below:



Notice that where two physical interface would have shown up, only a single logical interface (Po1) is seen and it is in the forwarding state.

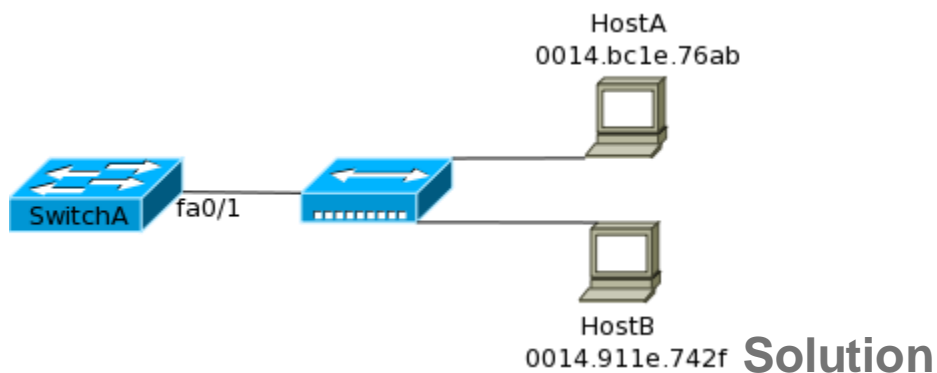
We will now shut down port 0/11 on SW2 and see the effect on the etherchannel and STP

In the above outputs notice that Po1 is still in forwarding mode and the trunk is still active. The status of a physical interface in an Etherchannel does not effect STP or Trunking. This is true as long as a single physical interface remains active in the etherchannel. If all physical interfaces go down, the channel will go down also and effect STP and Trunking.

## **6-8 Lab 6-1 – Port Security**

In the network shown in Figure 6-15, a hub has been connected to interface fa0/1 of SwitchA. It is an 8-port hub but only 5 hosts are allowed to connect to it. The administrator wants to ensure that only 5 hosts can connect. HostA and HostB along with any 3 other hosts can connect to the hub. Configure port security on switchport fa0/1 to fulfill this requirement. In case of a violation, the port should not be put in an error disabled mode but the administrator should be informed.

**Figure 6-15** Lab 6-1



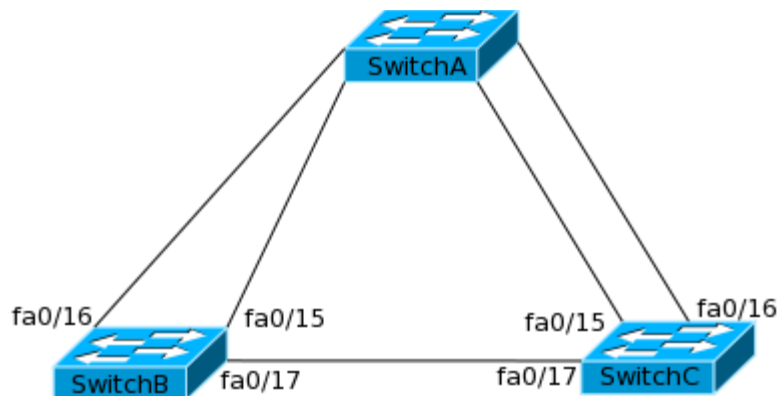
The lab requires configuring port security such that a maximum of 5 hosts can connect at a time. The MAC addresses of the two hosts also need to be added to port security and the violation mode must be changed to restrict. The configuration required is shown below:

You can verify the configuration using the **show port-security** command as shown below:

## **6-9 Lab 6-2 – STP**

In the network shown in Figure 6-16, 802.1d STP is being used on VLAN 5. The administrator wants to ensure that SwitchA is always the root bridge. They also want to ensure that interface fa0/16 is always the root port on SwitchB and SwitchC. In addition to that, interface fa0/1 on all switches should transition to forwarding as soon as a host connects to it.

**Figure 6-16** Lab 6-2



### **Solution**

To ensure that SwitchA is always the root bridge, change the priority on it as shown below:

To ensure that fa0/16 is always the root port on SwitchB and SwitchC, change the port costs as shown below:

Finally, to ensure that fa0/1 transitions to forwarding as soon as host connects, enable portfast on these ports as shown below:

To verify that SwitchA is the root bridge, use the **show spanning-tree vlan 5** command on SwitchA as shown below:

To verify that fa0/16 is the root port, use the show spanning-tree vlan 5 command on SwitchB and SwitchC as shown below:

## Summary

In this chapter you learned about the second most important layer covered in CCNA. This means that most of the topics covered in this chapter are very important from the exam perspective. From this chapter ensure that you remember:

- Three functions of a switch.
- Problems created by loops
- Root bridge election and how to influence them using bridge priority
- Root port election and how to influence them using port cost.
- Various versions of STP and Cisco proprietary changes to them.
- Switching methods.

STP is most of the most complex and important topics covered in CCNA. The next chapter covers another such topic so I strongly suggest you go through this chapter again before moving to the next one.

## **Chapter 7 VLANs and VTP**

The previous chapter introduced you to the world of LAN switching. We discussed switching loops and their negative impact on network performance while learning several techniques to prevent those loops using dynamic methods. We are now all set to explore the more interesting enhanced switching technologies including: VLANs, ISL and 802.1q trunking, VLAN trunking protocol (VTP), inter-VLAN routing, and Voice VLANs. By the time you finish this chapter, you will be armed with all the information you need to configure, verify, and troubleshoot these technologies on your Cisco Certified Network Associate (CCNA) exam as well as the real world. Please note that the examples in this chapter were created on Cisco 3550 and Cisco 3560 switches.

- 7-1 MAC Address Table
- 7-2 Virtual LANs (VLANs)
- 7-3 Types of Switch Ports
- 7-4 VLAN Trunking: ISL and 802.1Q
- 7-5 VLAN Trunking Protocol (VTP)
- 7-6 Inter-VLAN Routing
- 7-7 VLAN Configuration



- 7-8 Inter-VLAN Routing Configuration
- 7-9 VTP Troubleshooting
- 7-10 Voice VLAN Configuration

## **7-1 MAC Address Table**

The ultimate goal of switches is to carry frames from the source to the appropriate destination based on the destination Ethernet address in the frame header. Ethernet addresses, also known as MAC (Media Access Control) addresses, are 6 bytes or 48 bits in length, typically written in hexadecimal form. A Microsoft Windows system would list a MAC address as **12-34-56-78-9A-BC** whereas a Cisco switch would list it as **1234.5678.9abc**. They are merely different representations of the same MAC address.

Let's make a distinction between *frame* and *packet* before moving forward as these terms are often used rather loosely. The term *frame* refers to the bits and bytes that include the layer 2 header and trailer along with the data encapsulated by the header and trailer. The term *packet* is used to describe the layer 3 header and data without the layer 2 header or trailer.

The switch maintains an address table called MAC address table in order to efficiently switch frames between interfaces. When the switch receives a frame, it associates the MAC address of the sending device with the switch port on which it was received. In this

way, a switch dynamically builds an address table by using the source MAC address of the frames received.

As a practice, try issuing command **ipconfig /all** on Windows CLI and **show mac address-table** on Cisco switch CLI to get warmed up for later examples. The command **show mac address-table**, not surprisingly, is used to display the MAC address table of a switch we just talked about, as shown below:

In the above output, you should be able to identify various dynamically learned MAC addresses and the switch ports those MAC addresses are associated with. Each dynamically learned MAC address is associated with one and only one switch port. However, there may be more than one MAC addresses associated with the same switch port which means multiple devices are reachable off the same switch port. There are two possible scenarios in which multiple MAC addresses may be associated with the same switch port. In one case, the switch port may be connected to another switch which in turn has multiple devices connected to it. In the second case, multiple devices may be directly connected to the same switch port through a hub.

## **How Switches Work**

The basic logic used by switches when forwarding frames is the key to understanding many enhanced switching concepts and is worth a quick review here. The forwarding logic differs based on the type of destination MAC address and on whether the destination address has been added to the MAC address table of the switch.

### **Known Unicast**

The switch already has an entry in its MAC address table for the destination MAC address in the frame so it knows exactly which interface leads to the destination of the frame. The switch forwards frame out the single interface associated with the destination MAC address in the frame.

Figure 7-1 describes how known unicasts are propagated in a switched network. Host A sends a frame destined to host B which is forwarded by the intermediate switches to its final destination following the direction of the arrows.

**Figure 7-1** Known Unicast Propagation

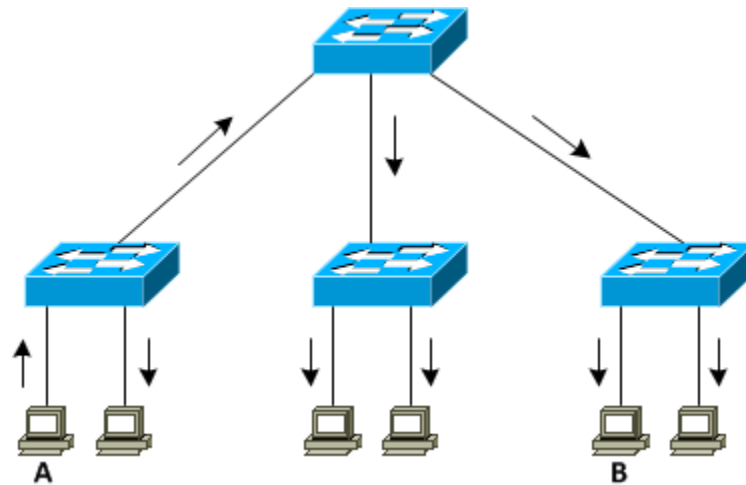
### Unknown Unicast

The switch has no entry in its MAC address table for the destination MAC address in the frame. The switch sends a copy of the frame out all interfaces, other than the interface on which the frame was received. The idea here is that the frame would ultimately reach all hosts and the host having the same MAC address as the destination address of the frame would accept it while all other hosts would reject the frame.

Figure 7-2 describes how unknown unicasts are propagated in a switched network. Host A yet again sends a frame destined to host B but this time intermediate switches do not yet have an entry for the MAC address of host B in their MAC address tables. Pay careful attention to the direction of arrows and note that the frame sent by host A is received by all hosts in the switched network including hosts connected to other switches. Only host A whose MAC address matches the destination MAC address of the frame would accept that frame while all other devices would reject it. It is not difficult to understand that a lot of bandwidth is wasted here because the frame is sent to every

host on the switched network. The negative impact would be more significant in a larger switched network with several switches and possibly hundreds of host.

**Figure 7-2** Unknown Unicast Propagation

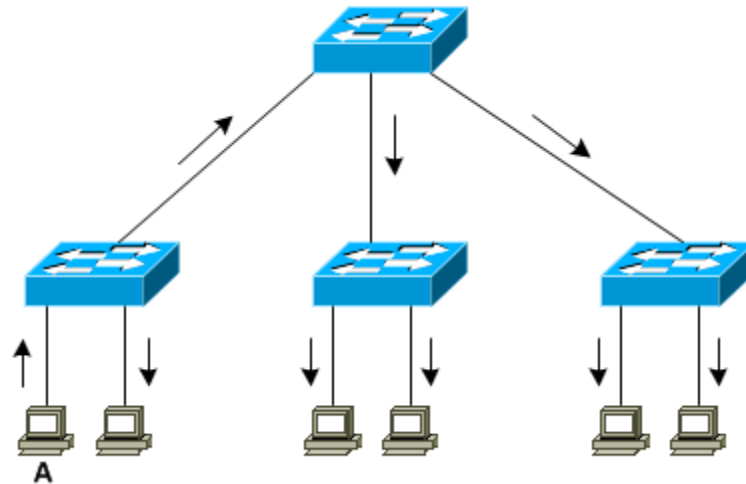


## Broadcast

The switch sends a copy of the frame out all interfaces, except the interface on which the frame was received, identically to unknown unicasts. This switch behavior is also called frame flooding.

Figure 7-3 describes how broadcasts are propagated in a switched network. Host A sends a broadcast frame with the broadcast destination MAC address of FFFF.FFFF.FFFF and the frame is propagated to all hosts in the network even those connected to other switches.

**Figure 7-3** Broadcast Propagation



## Multicast

The switch floods frame identically to unknown unicasts and broadcasts, unless certain multicast optimizations are configured.

There are some problems with the way switches forward different types of frames by default, especially in larger switched networks. First, there is no isolation between hosts and any host can communicate with any other host totally unchecked. This is not a very desirable situation for you as a network administrator as there is no security from MALICIOUS SOFTWARE or users. Second, a broadcast sent by any host would reach every other host on the network which is neither bandwidth efficient nor secure. A malfunctioning Network Interface Card (NIC) or a piece of malicious software on a host can generate excessive broadcasts consuming all the bandwidth available and starving legitimate applications. These problems can be greatly alleviated by using virtual LANs or VLANs.

## **7-2 Virtual LANs (VLANs)**

In order to appreciate the need for virtual LANs, let's consider how LANs would be built without switches using hubs only. As you are aware hubs are layer 1 devices without any intelligence and they typically relay the frame received on one port to all other ports regardless of the type of frame. As a matter of fact they don't care what the content of the frame is and the same treatment is given to unicast, multicast, and broadcast frames. As a result, the set of devices connected to a hub are in the same *collision domain* which means two devices connected to a hub cannot transmit at the same time without causing a collision. The Carrier Sense Multiple Access / Collision Detection (CSMA/CD) mechanism of Ethernet is at work in networks built with hubs. As all devices are in the same collision domain there is performance degradation as more and more devices are connected to the same hub and more collisions start to take place.

Let's assume we have five physical LANs in our organization: Engineering, Finance, Management, Marketing, and Sales each belonging to one department that need to be connected to the same router which provides Wide Area Network (WAN) connectivity. Here is how this network can be built using hubs alone.

**Figure 7-4** Physical LANs

Please note that enterprise networks do not use hubs any more and are built exclusively with switches. But analyzing the above network built with hubs would enable us to appreciate the benefits switches bring. First there is one hub for each physical LAN and all devices in that physical LAN are cabled to the same hub while each hub itself is connected to a separate interface on the router. Also, there is one IP subnet for each physical LAN and any device that is connected to that LAN has to have an IP address in that IP subnet. The router interface on a certain physical LAN also has an IP address in the IP subnet for that LAN. The hosts in the physical LAN have the router's IP address set as their default gateway. In this design, if you need to add another device in a physical LAN say Engineering, you simply connect it to the Engineering hub and assign it an IP address from the IP subnet for Engineering LAN.

There are some shortcomings in this design. First there are limitations on where you can physically place devices on a certain LAN due to the limited maximum cable length supported by Ethernet cabling standards. Let's assume there is a new employee in the Engineering department that needs to be connected to the Engineering LAN but there is no physical space in the Engineering department to make room for the new employee. There is plenty of space in the Sales department and the new Engineering employee is made to sit in the Sales department instead. Now, the Sales department is located in another corner of the building and it is not possible to connect the new Engineering employee to the Engineering hub due to the simple fact that the distance exceeds the maximum cable length permissible. The new Engineering employee is instead



connected to the Sales hub but this has some undesirable side effects. The Engineering employee is now on the Sales LAN and he can access all resources on the Sales LAN like servers which are meant to be visible only to the Sales people. It is a security issue as organization policies may prevent employees from other departments to have access to Sales documents and data. Also the new Engineering employee would be cut off from resources on the Engineering LAN which may prevent him from effectively doing his job. In the coming sections we will see how virtual LANs in networks built with switches instead of hubs provide means to prevent problems like this yet enabling physical mobility. The design I just described, though obsolete today, has worked well for several years despite its limitations.

In an Ethernet LAN, a set of devices that receive a broadcast sent by any other device is called a *broadcast domain*. We just learnt in the last section, a switch simply forwards all broadcasts out all interfaces, except the interface on which it received the frame. As a result, all the interfaces on an individual switch are in the same broadcast domain. Also, if a switch connects to other switches too, the interfaces on those switches are also in the same broadcast domain. On switches that have no concept of virtual LANs (VLANs), the whole switched network is one large flat network comprising a single broadcast domain.

A VLAN is simply a subset of switch ports that are configured to be in the same broadcast domain. Switch ports can be grouped into different VLANs on a single switch, and on multiple interconnected switches as well. By creating multiple VLANs, the switches create multiple broadcast domains. By doing so, a broadcast sent by a device in one VLAN is forwarded to all other devices in that same VLAN; however the broadcast is not forwarded to devices in the other VLANs. VLANs provide bandwidth efficiency because broadcasts, multicasts and unknown unicasts are restricted to individual VLANs and also provide security as a host on one VLAN cannot directly communicate with a host on another VLAN.



**Exam Concept** – A VLAN simply is a set of administratively defined switch ports that are in the same broadcast domain. The new CCNA exam has a ton of questions on VLAN concepts so understand them inside and out.

Because a trunk link can transport many VLANs, a switch must identify frames with their associated VLANs as they are sent and received over a trunk link. Frame identification assigns a unique user-defined number to each frame transported over a trunk link. This VLAN number is also called VLAN ID and as each frame is transported over a trunk link, such unique identifier is placed in the frame header. As each switch along the way receives these frames, the identifier is examined to determine to which VLAN the frames belong and then is removed. VLAN ID field contains a 15-bit value and as such the range of possible VLAN IDs is 0 – 4095. The VLAN IDs of 0 and 4095 are not used and the usable range of VLAN IDs hence is 1 – 4094. By default, all ports on a Cisco switch are assigned to VLAN 1. VLAN 1 is also called the management VLAN and control plane traffic belongs to VLAN 1.

Best practices recommended by Cisco dictate using a separate IP subnet for each VLAN. Simply put, devices in a single VLAN are typically also in the same IP subnet. Layer 2 switches forward frames between devices on the same VLAN, but they do not forward frames between devices in different VLANs. In order to be able to forward frames between two different VLANs, you need a multilayer switch or a router. We will cover this in more detail in a later section of the chapter.

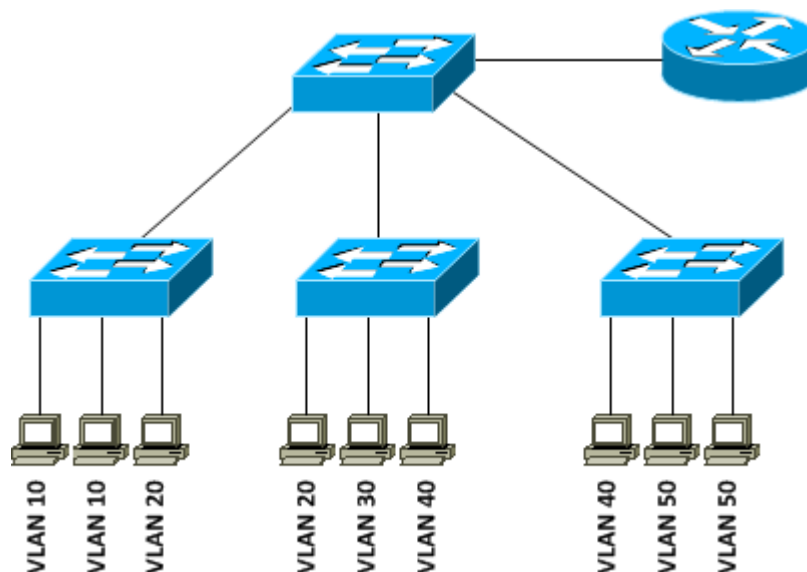
Now, let's see how we can build the same network using switches instead of hubs and what benefits switches bring to us. Table 7-1 lists VLAN IDs corresponding to organizational departments and IP subnets associated with each VLAN. There is a different IP subnet assigned to each VLAN and if you look carefully the third octet of the IP subnet number is the same as the VLAN ID. It is just an arbitrary number to make the design easily understandable and let us focus more on concepts rather than specific numbers used. The router has a Wide Area Network (WAN) connection and it provides two important functions: WAN connectivity and inter-VLAN routing to enable communication between two different departments or VLANs. As you can see in Figure 7-6, devices belonging to VLANs 20 and 40 are connected to more than one switch. VLANs remove the restriction of having to connect devices belonging to the same LAN

to the same device while still providing traffic isolation at layer 2. In yet other words, VLANs can span multiple switches with switch ports belonging to same VLAN existing on different switches. This is one of the several benefits switches bring to local area networks. Also if we want to add another device to say the Engineering VLAN and we want to locate the new user at a location different than the Engineering department, it can simply be accomplished by connecting the new device to the nearest switch and assigning the switch port to the Engineering VLAN. Compare it with the similar situation in the network built with hubs and you would be able to appreciate that network becomes more flexible without sacrificing security or traffic isolation. The router provides inter-VLAN routing in this scenario but the same functionality can also be achieved by using a layer 3 switch like the Cisco 3560.

**Table 7-1** VLANs Corresponding to Organizational Departments

VLAN IDs	Names	IP Subnet
10	Engineering	192.168.10.0/24
20	Finance	192.168.20.0/24
30	Management	192.168.30.0/24
40	Marketing	192.168.40.0/24
50	Sales	192.168.50.0/24

**Figure 7-6** Switches Using VLANs Remove Physical Boundaries



## VLAN Membership

One of the tasks that a system administrator has to perform while creating VLANs is to assign switch ports or interfaces to each VLAN. There are two ways switch interfaces can be assigned to VLANs and this gives rise to two different types of VLANs: static and dynamic. In the case of static VLANs, each switch port is statically assigned to a specific VLAN and any host connected to that switchport would automatically be a part of that VLAN. This kind of VLANs is static because individual switch ports are permanently allocated to specific VLANs. VLANs in this case are tied to switch ports and not to what is connected to those switch ports. In the case of dynamic VLANs however, all the host devices' hardware addresses are assigned into a database so the switch can be configured to assign VLANs dynamically any time a host is connected to a switch. In this case VLAN is tied to the hardware addresses or MAC addresses of hosts and not to switch ports. A host whose MAC address is tied to a certain VLAN would be part of that VLAN regardless of which switch port it is connected to. Static VLANs are easier to create than dynamic VLANs because there is no need to document the hardware addresses of all hosts that would possibly be connected to the LAN and then to store them in a database on the switch. I will cover both static and dynamic VLANs in the coming sections.

### **Static VLANs**

The most common and straightforward method to create VLANs is static VLANs and these are also more secure than dynamic VLANs. The security hinges on the fact that VLANs are manually associated to switch ports by switch configuration and these VLAN associations are always maintained unless the port assignment is manually changed.

Static VLAN configuration is pretty easy and it works really well in most enterprise networking environments where user mobility is limited and controlled. End points typically consist of desktop computers, printers and servers which are permanently cabled to switch ports. It is fitting to associate VLANs to switch ports in such an environment.

### **Dynamic VLANs**

Dynamic VLAN bases VLAN assignment on hardware (MAC) addresses, protocols, or even applications that create dynamic VLANs. Let's consider a specific case where MAC addresses have been entered into a centralized VLAN management application and you connect a new node to a switch. If the node is attached to an unassigned switch port, the VLAN management database can look up the hardware address and both assign and configure the switch port into the correct VLAN. This makes configuration and management a lot easier because if a user moves to a new location and plugs the node into a new switch port, the switch simply will assign them to the correct VLAN automatically. It is possible because VLAN assignment is tied to hardware addresses of nodes rather than physical switch ports. But this ease of VLAN management and user mobility comes at a cost: you have to do a lot more work initially noting the hardware addresses of all nodes that are supposed to be connected to the LAN and set up the VLAN database accordingly.



**Exam Concept –** Control plane traffic such as VTP, CDP, DTP, and PAgP protocols is always sent in VLAN 1 across a trunk link between two Cisco switches.

## **7-3 Types of Switch Ports**

A switch port can be in one of two modes: *access* and *trunk*. There are two ways a switch port can settle down into one of these two modes: static and dynamic. You can manually configure a switch port to be in the access or trunk mode in the static method. You can also let Dynamic Trunking Protocol (DTP) run on an interface to negotiate trunking in the dynamic method. Cisco switches exchange DTP messages to dynamically learn whether the device at the other end of the link wants to perform trunking and, if so, which trunking protocol (ISL or 802.1Q) to use.

### **Access Ports**

A switch port in access modes belongs to one specific VLAN and sends and receives regular Ethernet frames in untagged form. The switch interfaces connected to devices such as desktops, laptops, printers etc. are typically configured as access ports. By default, a Cisco switch port is assigned to the default VLAN 1 in access mode. You can explicitly set the switch port to access mode using command **switchport mode access** in interface configuration mode. The VLAN that certain switch port is assigned

to can be changed using command **switchport access vlan** *vlan-id*, in interface configuration mode.

We just configured interface FastEthernet 0/1 of switch SW1 in access mode assigning it to VLAN 10.

### Trunk Ports

The distinguishing feature of trunk ports is that they carry traffic from multiple VLANs at the same time. Such interfaces are most commonly configured between two switches but they can also be configured between a switch and a router, and even between a server and a switch. The range of VLAN IDs that can be configured on a Cisco switch is 1 to 4094 which is divided into normal-range VLAN IDs of 1 to 1005 and extended-range VLAN IDs of 1006 to 4094.

In fact trunking is a great feature because a single physical link is shared by multiple VLANs while still allowing traffic isolation between different VLANs. In the absence of such feature we would have required one inter-switch link per VLAN which would simply not scale to a large number of VLANs. By default the full range of VLAN IDs 1 to 4094 is

allowed on a trunk port which means traffic belonging to all VLANs can be carried across the trunk port. It is also possible to allow only a subset of the full range of VLAN IDs on the trunk while blocking the others. Trunking allows a VLAN to span multiple switches with access ports belonging to the VLAN spread across multiple switches in different parts of the switched network. This provides great flexibility when creating VLANs and a host can be assigned to a VLAN regardless of its physical location on the switched network.

**Exam Concept** – A trunk link must operate at 100 Mbps or greater speeds. This is a common CCNA question.

A switch port can be configured as trunk using command **switchport mode trunk** in interface configuration mode.

We will learn more about trunking protocols ISL and 802.1Q in a later section.

### Voice Access Ports

Voice access ports are a special case of access ports with modified behavior suited for connecting IP phones. Most corporate users these days use two network devices: a desktop or laptop computer and an IP phone. Typically just one LAN cable runs from the desk or cubicle to the switch that carries data traffic from the computer and voice traffic from the IP phone. Voice access ports allow you to add a second VLAN to an access port on a switch for your voice traffic which is called the voice VLAN. In fact a voice access port is like a hybrid of an access port and a trunk port carrying some characteristics of each type, but it is still considered an access port that can be



configured for both data and voice VLANs. So what we get is the ability to use the same physical interface and the same physical cable run for both data and voice traffic yet compartmentalizing each type of traffic in its own VLAN.

## **7-4 VLAN Trunking: ISL and 802.1Q**

As you learned, VLAN trunking allows switches to send Ethernet frames for multiple VLANs across a single link. A trunk interface needs a way to distinguish between Ethernet frames that belong to different VLANs. If frames from different VLANs are sent unaltered across the trunk interface, the switch at the other end would have no way of knowing which VLAN certain frame belongs to. This leads us naturally to the idea of frame tagging. Frame tagging is simply adding some additional information to regular frames before sending them out a trunk interface so that the device at the other end of the trunk interface would identify the VLAN the frame belongs to.

VLAN IDs are associated with only those frames that traverse a trunk link. When a frame enters or exits the switch on an access switch port, no VLAN ID is present. The Application Specific Integrated Circuits (ASICs) on the switch port physically assign the VLAN ID to a frame as it is placed on a trunk link and also strips off the VLAN ID if the frame exits an access switch port. When we speak of ASICs we are in the realm of

hardware architecture of the switch, performing frame tagging in hardware which allows us to match wire speeds.

There are two different ways to tag frames: ISL and 802.1Q. Although the basic concept of frame tagging is the same with both methods, there are differences in how it is accomplished. If two devices are to perform trunking, they must agree to use either ISL or 802.1Q as there are several differences between the two.

**Table 7-2** Comparison of ISL and 802.1Q

<b>Feature</b>	<b>ISL</b>	<b>802.1Q</b>
Supported VLANs	Normal and extended range	Normal and extended range
Protocol defined by	Cisco	IEEE
Encapsulates original frame or inserts tag	Encapsulates	Inserts tag
Native VLAN support	No	Yes

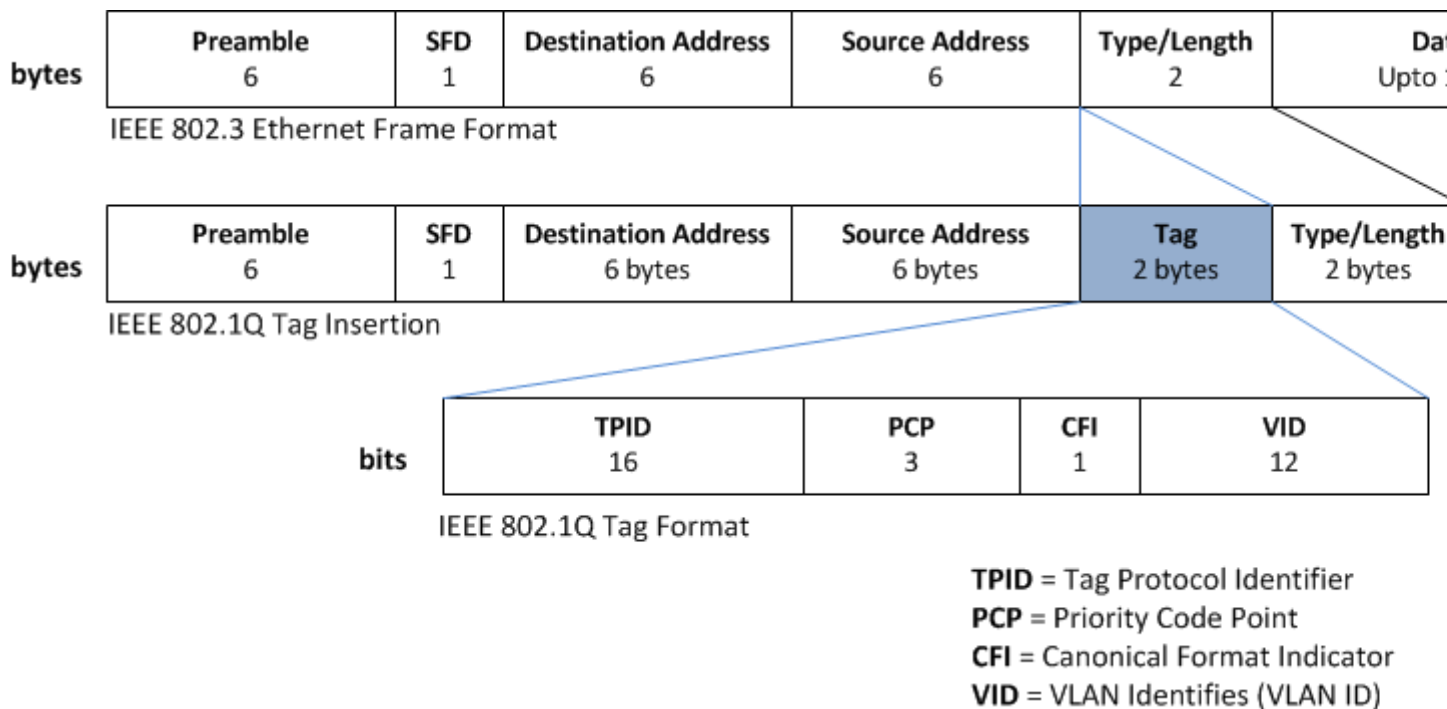
### **ISL and 802.1Q Concepts**

Inter-switch Link (ISL) is a Cisco proprietary protocol that maintains VLAN information in Ethernet frames by encapsulating the whole Ethernet frame. In the case of ISL, the tag is external to the Ethernet frame, which is the same as encapsulating the Ethernet frame. ISL adds a 26-byte header (containing a 15-bit VLAN identifier) and a 4-byte CRC trailer to the frame. ISL is supported only on Cisco switches and even some newer Cisco switches don't support it any more. ISL cannot be used to connect a Cisco switch to a switch by another vendor like HP and its use is being depreciated even by Cisco in favor of IEEE 802.1q which happens to be the more popular choice among trunking protocols.

IEEE 802.1q is a standard developed by the Institute of Electrical and Electronics Engineers (IEEE) to carry traffic belonging to multiple VLANs across a trunk. In contrast to ISL, 802.1Q does not actually encapsulate the original frame. Instead, it adds a 32-bit field between the source MAC address and the Ether Type/Length fields of the original frame. This 32-bit field carries the information used to deterministically identify the VLAN the Ethernet frame belongs to.

The extra VLAN header used by both ISL and 802.1Q uses the *VLAN identifier* or *VLAN ID* field to identify the VLAN the frame belongs to. VLAN ID is a 12-bit field specifying the VLAN to which the frame belongs. The range of hexadecimal values is from 0x000 to 0xFFF for a 12-bit number. The hexadecimal values of 0x000 and 0xFFF are reserved while all other values in the range can be used as VLAN identifiers, allowing up to 4,094 VLANs. Please see the graphic to understand how IEEE 802.1Q tag is inserted in a regular Ethernet frame.

**Figure 7-7** IEEE 802.1Q Tag Insertion



The IEEE 802.1Q standard can create a very interesting scenario with Ethernet frames of maximum size. Please recall that the maximum size of an Ethernet frame is 1518 bytes as specified by IEEE 802.3 standard. Now, if such frame gets tagged the resulting frame size will be 1522 bytes, a number that exceeds the maximum size specified in IEEE 802.3 standard. In order to resolve this issue the maximum Ethernet frame size was extended to 1522 bytes by the **802.3ac** subgroup of the IEEE 802.3 committee. Still some network devices that do not support the larger frame size will process the frame successfully but may report these larger frames as *baby giant*.

IEEE 802.1Q and ISL are used to multiplex VLANs over single link by adding VLAN tags for identification. However, it is possible to send Ethernet frames either tagged or untagged across an IEEE 802.1Q trunk. Cisco uses the concept of native VLAN to help explain which frames will be sent with or without tags. An IEEE 802.1Q trunk port sends and receives tagged frames for all VLANs, except the native VLAN if one is configured. Frames belonging to the native VLAN do NOT carry VLAN tags when sent over the trunk. Similarly, if an untagged frame is received on a trunk port, the frame is associated with the native VLAN configured on that port. The concept of native VLAN is not important for ISL as all frames including the ones for native VLAN are tagged. The default native VLAN on Cisco switches is 1. Also please note that the native VLAN is specific to a single trunk port and not to the whole switch. In fact different trunk ports on a Cisco switch can have different native VLANs. Both the trunk ports at the two ends of a trunk should have the same native VLAN configured.

On a side note, many Network Interface Cards (NICs) for PCs and printers are not 802.1Q compliant. If they receive a tagged frame, they will not understand the VLAN tag and will drop the frame. From a practical standpoint, a PC should get one and only one VLAN so it does not matter if your PC NIC supports dot1Q or not. However NICs on server machines may support 802.1Q and there are situations where this capability is useful. You may provide access to applications on server to different VLANs still providing traffic isolation. As the server NIC is 802.1Q capable it can receive traffic from different VLANs on the same physical interface by establishing an 802.1Q trunk link with the switch it is directly connected to.

## **7-5 VLAN Trunking Protocol (VTP)**

VLAN Trunking Protocol (VTP) was developed by Cisco to reduce VLAN administration effort in a switched network, making it a Cisco proprietary protocol. The comparable IEEE standard in use by other manufacturers is *GARP VLAN Registration Protocol (GVRP)*, and more recently *Multiple VLAN Registration Protocol (MVRP)*. As you know by now, there are two important tasks to be performed when creating VLANs in a switched network: creating VLANs and assigning switch ports to VLANs. The first task requires the network administrator to define all the VLANs on each switch in a switched network. If performed manually by logging into each switch, this can be a tedious task on a large network involving a large number of switches and is also prone to error. VLANs can be created on only a single switch and this VLAN information is propagated through VTP messages to all switches in the network. This not only greatly reduces the effort involved but also minimizes the chance of an error. VTP allows you to add, delete, and rename VLANs on a single switch and this information is then propagated to all other switches in the VTP domain.

On a side note, the name VLAN Trunking Protocol (VTP) may be a bit misleading as the protocol does not have much to do with trunking. VTP just makes it easier to define VLANs by doing it on one central switch and propagating that information to the whole switched network through VTP messages. In this manner, VTP allows for more consistent VLAN configuration, and accurate tracking and monitoring of VLANs by central administration. In other words, a switch can only share VLAN information with other switches over VTP if they are configured into the same VTP domain. VTP

information is sent only over trunk ports whereas no VTP information is sent over access ports. Switches not only advertise all known VLANs with any specific parameters but also VTP management domain information and configuration revision number.

### VTP Modes of Operation

A switch can operate in one of three different modes of operation within a VTP domain:

**Server** This is the default mode on all Cisco Catalyst switches. The switch in VTP server mode is needed to propagate VLAN information throughout the VTP domain. Also, a switch must be in VTP server mode to be able to create, modify, and delete VLANs. VTP information should be changed on the switch operating in server mode and any change made to a switch in server mode will be propagated throughout the VTP domain via VTP advertisements forwarded on trunks. Also, VLAN configurations are saved in NVRAM for switch in VTP server mode.

**Client** A switch in VTP client mode receives information from VTP servers, but it also sends and receives VTP updates just like VTP servers. But, in contrast to VTP server, a VTP client cannot create, modify, or delete VLANs. Also, you cannot assign a port on a VTP client to a VLAN before the VTP server notifies the client of the new VLAN. Also, a VTP client does *not* store the VLAN information it receives from a VTP server in NVRAM. This means that if the switch loses power or is reloaded, the VLAN information it has learnt would be gone and it would have to re-learn the information from a VTP server. So basically, switches that are in VTP client mode will just learn and pass along VTP information.

**Transparent** Switches in VTP transparent mode receive VTP advertisements and forwards them over any configured trunk links, but that's all. They do not update their own VLAN database with the VTP information they receive and pass along. Also, they can create, modify, and delete VLANs in their own VLAN database but this database is kept isolated from the rest of the VTP domain and is not advertised at all. Practically, switches in VTP transparent mode do not participate in the VTP domain and act just as relay agents receiving VTP advertisements and passing them along. The utility of VTP

transparent mode is to enable VTP servers and clients synchronise their VLAN databases even if they are connected via switches that are not supposed to have the same VLANs.

A switch can be configured in VTP transparent mode to receive and forward VTP information through trunk ports but not to update their VLAN database. In other words, switches in transparent mode only relay VTP information without updating their own VLAN databases.

**Exam Concept** Typically you will see questions on the CCNA exam about VTP modes. Know that a switch has to be in VTP server or transparent mode in order to make any VLAN changes locally.

**VTP Domains** Cisco switches participating in VLAN Trunking Protocol (VTP) are organized into management domains, or areas with similar VLAN requirements. A switch can be part of one and only one VTP domain and can share VLAN information with other switches in the same domain. Switches in different VTP domains do not share VTP information. If a switch receives a VTP advertisement from a switch in a different VTP domain, it will ignore such advertisement. Mismatched VTP domain names are a common cause why all switches in your network do not share VLAN information and should be one of the first things you should check when troubleshooting VTP issues.

The concept of a VTP management domain is somewhat analogous to the concept of autonomous system (AS) in Border Gateway Protocol (BGP). A switch can belong to only one VTP domain just like a BGP router can belong to a single AS.

**Exam Concept –** You will see a CCNA exam question asking what happens if a switch receives a VTP advertisement with a different management domain name. Know it simply ignores such an advertisement.

Switches in a VTP domain advertise several attributes to their VTP domain neighbors. Each advertisement contains several parameters including VTP management domain, VTP revision number, known VLANs, and specific VLAN parameters. When a new VLAN is added to a switch in a VTP domain, other switches are notified of the new VLAN through VTP advertisements. In this way

## **7-6 Inter-VLAN Routing**

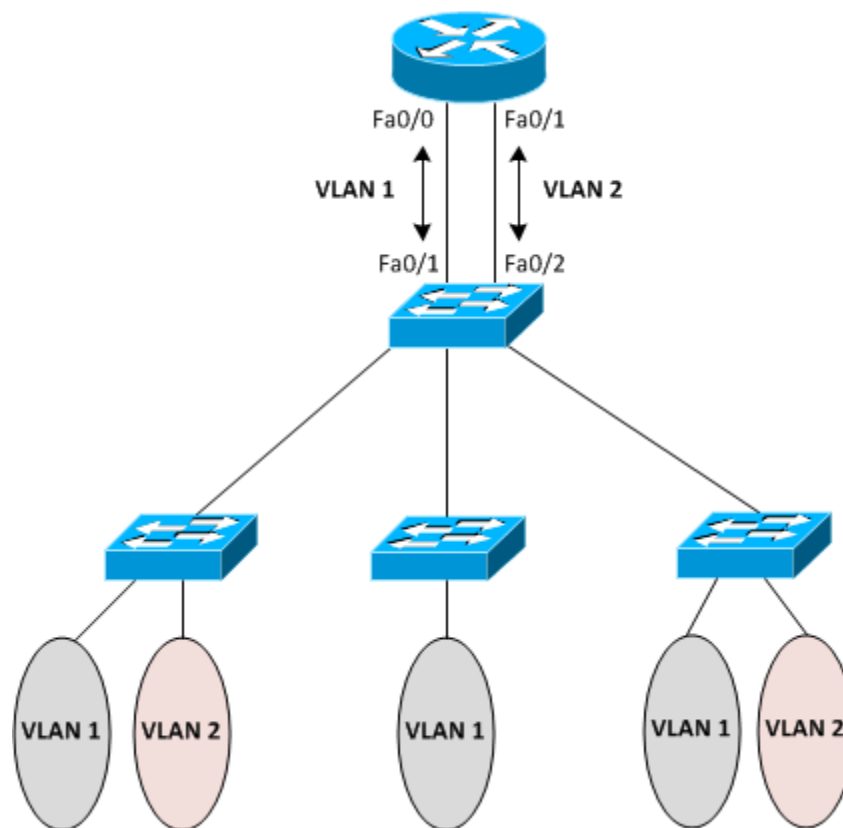
VLANs provide traffic separation at layer 2 of the OSI model. Hosts in a VLAN can communicate freely and directly with other hosts on the same VLAN and it includes unicasts, multicasts, and broadcasts. All three types of frames can flow freely and directly between any two hosts that are on the same VLAN regardless of their physical location on a switched network. But what if hosts on two different VLANs need to communicate? In such situation you need a layer 3 device, either a router or a layer 3 switch. Such communication is simply not possible within the bounds of a layer 2 only network.

Have a look at Figure 7-7, where our switched network has two VLANs: VLAN 1 and VLAN 2. Hosts in VLAN 1 need to communicate with hosts in VLAN 2. We know that this kind of communication is not possible in our layer 2 only switched network and we need a layer 3 device. One possible solution to achieve communication between the two VLANs can be to introduce a router into the picture such that the router has two LAN interfaces Fa0/0 and Fa0/1 one for each VLAN. These two interfaces are connected to two access switch ports Fa0/1 and Fa0/2 in VLANs 1 and 2 respectively. The router interfaces connected to these switch ports each have an IP address configured in the subnet corresponding to the associated VLAN. From the standpoint of the router, the two VLANs are merely two different subnets connected to two different



router interfaces and the router essentially performs routing to move traffic between the two VLANs. Please remember that best practices dictate using a separate IP subnet for each VLAN. As you can see, this scheme requires one dedicated interface on the router for each VLAN in your switched network. You can imagine the solution does not scale well when you have several VLANs in your switched network.

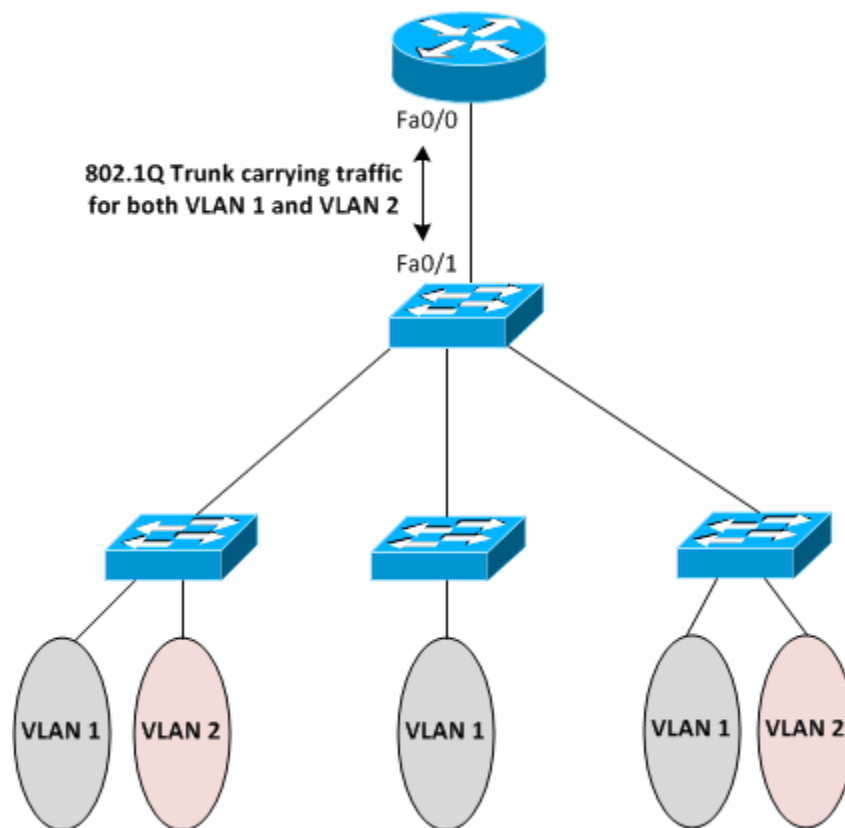
**Figure 7-8** Router with Separate Interface for Each VLAN



Now, have a look at Figure 7-8 below, which is an alternate and more efficient way of achieving routing between different VLANs using a router. Here we have only one router interface Fa0/0 connected to the switch port Fa0/1. The link is configured as an 802.1Q carrying traffic for both VLAN 1 and VLAN 2. There is one sub-interface per VLAN configured on the router with IP addresses configured on subinterfaces rather than the physical interface. This is the key difference that we have only one physical connection from the router to the switch regardless of the number of VLANs. This solution, also called router-on-a-stick, is more efficient and scalable to a large number of VLANs. Most

switches today are not just layer 2 devices but are multilayer switches and inter-VLAN routing can be achieved using switches alone without involving a router at all. You will see this concept on the CCNA exam and you must remember that the link from the switch to the router must be a trunked link and the router's interface must be at least a Fast Ethernet interface. These two things very important and will be on the exam!

**Figure 7-9** Router on a Stick



## **7-7 VLAN Configuration**

VLAN concepts may be a bit overwhelming at first, but surprisingly the actual configuration of VLANs in a network of Cisco switches requires just a few simple steps:

**Step 1** Create the VLAN.

**Step 2** Assign switch ports to that VLAN.

In Example 7-2, we will create several new VLANs on Switch1 and also assign names to them according to Table 7-1.

**Table 7-2** VLANs to be created

<b>VLAN IDs</b>	<b>Names</b>
10	Engineering
20	Finance
30	Management
40	Marketing
50	Sales

We start out by displaying existing VLANs on the switch using command **show vlan brief**. The switch displays VLANs 1, 1002, 1003, 1004, and 1005 which are created by default on Cisco switches and cannot be removed. As we have not created any new VLANs yet this output is just what we expect.

## TUN MIN OO {BE-IT} Routing & Switching 200-120

```
Switch1>enable
Switch1#show vlan brief
```

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/2, Fa0/3 Fa0/4, Fa0/5, Fa0/6 Fa0/7, Fa0/8, Fa0/9 Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15 Fa0/22, Fa0/23, Fa0/24 Gi0/1, Gi0/2
1002 <u>fddi</u> -default	act/ <u>unsup</u>	
1003 <u>token-ring</u> -default	act/ <u>unsup</u>	
1004 <u>fddinet</u> -default	act/ <u>unsup</u>	
1005 <u>trnet</u> -default	act/ <u>unsup</u>	

Now we create new VLAN IDs 10, 20, 30, 40, and 50 with names Engineering, Finance, Management, Marketing, and Sales respectively. Please note that we have assigned names to VLANs but this step is optional. If names are not explicitly assigned to VLANs, a Cisco switch automatically creates a VLAN name for each VLAN created which is drawn from the VLAN ID itself.

Now we display VLANs again just like we did at the start to verify that new VLANs have been created using the show vlan brief command.

```
Switch1#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/3 Fa0/4, Fa0/5, Fa0/6 Fa0/7, Fa0/8, Fa0/9 Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15 Fa0/22, Fa0/23, Fa0/24 Gi0/1, Gi0/2
10	Engineering	active	
20	Finance	active	
30	Management	active	
40	Marketing	active	
50	Sales	active	
1002	<u>fddi</u> -default	act/ <u>unsup</u>	
1003	<u>token-ring</u> -default	act/ <u>unsup</u>	
1004	<u>fddinet</u> -default	act/ <u>unsup</u>	
1005	<u>trnet</u> -default	act/ <u>unsup</u>	

You can see that five new VLANs have been successfully created but also note carefully in the Ports column that now switch ports are yet assigned to these newly created LANs.

Now we will proceed to assign switch ports to VLANs.

**Table 7-3** Ports to be Assigned to VLANs

VLAN IDs	Ports
10	Fa0/1, Fa0/2, Fa0/3
20	Fa0/4, Fa0/5
30	Fa0/6
40	Fa0/7, Fa0/8
50	Fa0/9, Fa0/10

We are going to assign switch ports FastEthernet 0/1 to FastEthernet 0/3 to VLAN 10.

Now assign switch ports FastEthernet 0/4 and FastEthernet 0/5 to VLAN 20.

We will continue by assigning port FastEthernet 0/6 to VLAN 30.

Let's assign switch ports FastEthernet 0/7 and FastEthernet 0/8 to VLAN 40.

Now assign switch ports FastEthernet 0/9 and FastEthernet 0/10 to VLAN 50.

The command **show vlan brief** is turning out to be really useful we can use it yet again here to verify that switch ports have been assigned to VLANs as expected.

```
Switch1#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Fa0/11, Fa0/12, Fa0/14 Fa0/15, Fa0/17, Fa0/18 Fa0/20, Fa0/21, Fa0/22 Fa0/23, Fa0/24, Gi0/1 Gi0/2
10	Engineering	active	Fa0/1, Fa0/2, Fa0/3
20	Finance	active	Fa0/4, Fa0/5
30	Management	active	Fa0/6
40	Marketing	active	Fa0/7, Fa0/8
50	Sales	active	Fa0/9, Fa0/10
1002	<u>fddi</u> -default	act/ <u>unsup</u>	
1003	<u>token</u> -ring-default	act/ <u>unsup</u>	
1004	<u>fddi</u> net-default	act/ <u>unsup</u>	
1005	<u>trnet</u> -default	act/ <u>unsup</u>	

This completes our configuration and verification for this section but you may have noticed that there is quite a bit of repetition when it comes to assigning several switch ports to the same VLAN. There is a shortcut Cisco IOS provides to accomplish this with fewer commands by applying those commands to a range of switch ports. Lets again assign switch ports FastEthernet 0/1 to FastEthernet 0/3 to VLAN 10 using the new method.

As you can see it greatly reduces the effort needed to apply the same configuration to multiple switch ports. This method can be used to easily apply any commands from the interface configuration mode to a range of interfaces.

Please keep in mind that Cisco IOS switches keep VTP and VLAN information in a file named **vlan.dat** which is stored in the flash memory. Even if you erase the startup configuration and reload the device, VLAN information persists because it is saved in **vlan.dat** file. You must manually delete the vlan.dat file in addition to erasing the startup configuration if you want to get rid of all VLAN information on the switch.



**Real World Concept** – Before deleting a VLAN, re-assign the ports belonging to that VLAN to another VLAN in order to avoid making the ports inoperable.

### VLAN Management Policy Server (VMPS)

Cisco switches also support a dynamic method of assigning devices to VLANs, based on the device's MAC addresses, using a tool called VLAN Management Policy Server (VMPS). A VLAN Management Policy Server or VMPS is simply a Cisco switch that maintains device information to VLAN mapping. With VMPS, a switch administrator can dynamically assign a network device to a particular VLAN. This technology ties VLAN



membership to the end device rather than the switch port and is useful in sites that contain a large number of mobile users.

You can use the VLAN Management Policy Server (VMPS) service to set up a database of MAC addresses to be used for the dynamic addressing of your VLANs. The VMPS database automatically maps MAC addresses to VLANs. A dynamic access port can belong to one VLAN anywhere in the range 1-4094 and is dynamically assigned by the VMPS. Lower end switches like the Catalyst 2960 can be a VMPS client only.

### Trunk Port Configuration

You can manually configure trunk links on Cisco switches but Cisco has also implemented a proprietary, point-to-point protocol called Dynamic Trunking Protocol (DTP) that negotiates a common trunking mode between two neighboring switches. The negotiation covers the encapsulation (ISL or 802.1Q) and whether the link becomes a trunk at all. This allows trunk links to be used without much manual configuration or administration.

Now that you understand the two types of trunk interfaces, let's see how to configure each type. The following list describes the different options available to you when configuring a switch interface:

**switchport mode access** This command entered in interface configuration mode puts the interface into permanent non trunking mode and also negotiates to convert the link into a non trunk link. The interface becomes a non-trunk interface regardless of whether the neighboring interface is also a non-trunk interface. Such interface would be a dedicated layer 2 interface.

**switchport mode dynamic auto** This interface configuration mode command makes the interface able to convert the link to a trunk link dynamically only if the neighboring switch initiates DTP negotiation. The interface becomes a trunk interface if the neighboring interface is set to **trunk** or **dynamic desirable** mode. This is also the factory default mode on Cisco switch interfaces.

**switchport mode dynamic desirable** This interface configuration mode command makes the interface able to convert the link to a trunk link dynamically by actively initiating DTP negotiation. The interface becomes a trunk interface if the neighboring interface is set to **trunk**, **dynamic desirable**, or **dynamic auto** mode.

**switchport mode dynamic desirable** This command puts the interface into permanent trunking mode also negotiating to convert the neighboring interface into trunking mode. The interface becomes a trunk interface even if the neighboring interface is not a trunk interface.

**switchport nonegotiate** This interface configuration mode command prevents the interface from generating DTP frames to negotiate trunking. You can use this command only when the interface is configured with **switchport mode trunk** or **switchport mode access**. This command is not compatible with **dynamic auto** and **dynamic desirable** modes.

Dynamic Trunking Protocol (DTP) is not only used to negotiate trunking on a link between two devices but also to negotiate the encapsulation type of either 802.1Q or ISL. When we decide to make a link access or trunk using relative configuration commands, it is a good practice to disable DTP on the link by using switchport nonegotiate command to prevent unnecessary DTP traffic on the link.

**Table 7-4** Trunk Configuration Options

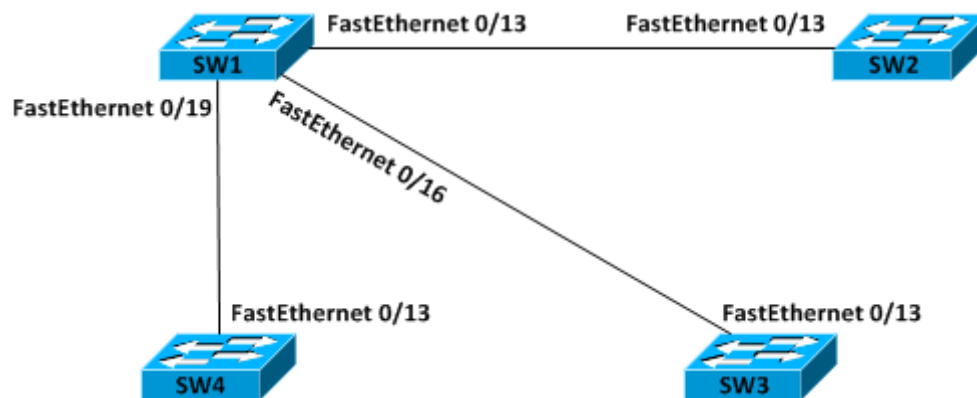
Configuration Command	Short Name	Meaning	Configuration on Other Side to Trunk
<b>switchport mode trunk</b>	Trunk	Always trunks; sends DTP messages to help other side choose to trunk	On, desirable, auto
<b>switchport mode trunk;switchport nonegotiate</b>	Trunk (with nonegotiate)	Always trunks; does not send DTP messages	On
<b>switchport mode dynamic desirable</b>	Desirable	Sends DTP messages and trunks if negotiation succeeds	On, desirable, auto
<b>switchport mode dynamic auto</b>	Auto	Replies to DTP messages and trunks if negotiation	On, desirable

		succeeds	
<b>switchport mode access</b>	Access	Never trunks; sends DTP messages to help the other side reach the same conclusion	Never trunks
<b>switchport mode access;switchport nonegotiate</b>	Access (with nonegotiate)	Never trunks; does not send DTP messages	Never trunks

Please see below diagram where SW1 is connected to SW2, SW3, and SW4 via interface Fa0/13, Fa0/16, and Fa0/19 respectively. Also note that SW2, SW3, and SW4 each has its interface Fa0/13 connected to an interface on SW1. This is how we are going to configure our switched network here:

- SW1 – SW2 configured as ISL trunk
- SW1 – SW3 configured as 802.1Q trunk
- SW1 – SW4 configured dynamically by DTP

**Figure 7-10** Trunking Configuration Reference for Example



We start out by configuring interface FastEthernet 0/13 of SW1 as trunk and setting the encapsulation to ISL.

We now configure interface FastEthernet 0/16 of SW1 as trunk and set the encapsulation to 802.1Q.

We will not change the default configuration on interface FastEthernet 0/19 of SW1 and let Dynamic Trunking Protocol (DTP) negotiate trunking with interface FastEthernet0/13 of SW4.

Now we move to SW2 and configure interface FastEthernet 0/13 as trunk and set the encapsulation to ISL.

Now we move to SW3 and configure interface FastEthernet0/13 as trunk and set the encapsulation to 802.1Q

We will not change the default configuration of interface FastEthernet0/13 of SW4 and let it negotiate trunking with interface FastEthernet0/19 of SW1 by Dynamic Trunking Protocol (DTP).

Now that our trunking configuration is complete on all switches, let's move to SW1 and see if trunks have been formed as expected.

```
SW1#show interface trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/13	on	isl	trunking	1
Fa0/16	on	802.1q	trunking	1
Fa0/19	auto	n-isl	trunking	1

Port	Vlans allowed on trunk
Fa0/13	1-4094
Fa0/16	1-4094
Fa0/19	1-4094

Port	Vlans allowed and active in management domain
Fa0/13	1
Fa0/16	1
Fa0/19	1

Port	Vlans in spanning tree forwarding state and not pruned
Fa0/13	1
Fa0/16	1
Fa0/19	1

In the above output of command **show interface trunk** on SW1, you can see three interfaces listed in *Port* column as expected. You can see in *Mode* column that Fa0/13 and Fa0/16 are in *on* mode while Fa0/19 is in *auto* mode. This is consistent with the fact that Fa0/13 and Fa0/16 are manually configured as trunks while Fa0/19 has DTP running on it. We move on to the *Encapsulation* column now where you see the encapsulation for Fa0/13 and Fa0/16 is *isl* and *802.1q* respectively. The encapsulation for Fa0/19 is listed as *n-isl* or negotiated ISL. This means DTP was running on this interface and the negotiation resulted in formation of a trunk with ISL encapsulation.

We move to SW2 now and see how trunks have formed.

## TUN MIN OO {BE-IT} Routing & Switching 200-120

```
SW1#show interface trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/13	on	isl	trunking	1
Fa0/16	on	802.1q	trunking	1
Fa0/19	auto	n-isl	trunking	1

Port	Vlans allowed on trunk
Fa0/13	1-4094
Fa0/16	1-4094
Fa0/19	1-4094

Port	Vlans allowed and active in management domain
Fa0/13	1
Fa0/16	1
Fa0/19	1

We move now to SW3 and verify trunking.

```
SW3#show interface trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/13	on	802.1q	trunking	1

Port	Vlans allowed on trunk
Fa0/13	1-4094

Port	Vlans allowed and active in management domain
Fa0/13	1

Port	Vlans in spanning tree forwarding state and not pruned
Fa0/13	1

Finally, let's move to SW2 and see how trunks have formed.

```
SW2#show interface trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/13	auto	n-isl	trunking	1

Port	Vlans allowed on trunk
Fa0/13	1-4094

Port	Vlans allowed and active in management domain
Fa0/13	1

Port	Vlans in spanning tree forwarding state and not pruned
Fa0/13	1

That was one doozy of an example, but we learnt how to create and verify trunking using different methods. There is some additional fine tuning that can be done to the trunk links as shown below:

### Defining the Allowed VLANs on a Trunk

By default the full range of VLANs 1 to 4094 are allowed on a trunk link. But you can selectively allow VLANs on a trunk while disallowing others using command **switchport trunk allowed vlan:**

The above command will only allow VLANs 1,10, 20, 30, 40 and 50 on the trunk while disallowing all others. The configuration can be verified using command **show interface trunk:**

```
SW1#show interface trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/13	on	isl	trunking	1
Fa0/16	on	802.1q	trunking	1
Fa0/19	auto	n-isl	trunking	1

Port	Vlans allowed on trunk
Fa0/13	1-4094
Fa0/16	1,10,20,30,40,50
Fa0/19	1-4094

Port	Vlans allowed and active in management domain
Fa0/13	1
Fa0/16	1
Fa0/19	1

Port	Vlans in spanning tree forwarding state and not pruned
Fa0/13	1
Fa0/16	1
Fa0/19	1

## Modifying the Trunk Native VLAN

The native VLAN is the one VLAN whose frames are not tagged with 802.1Q encapsulation before sending out an 802.1Q trunk. The native VLAN should match on both ends of a trunk link because the receiving end would interpret any frame received untagged on an 802.1Q trunk as belonging to the native VLAN. You can change the native VLAN using command **switchport trunk native vlan**.

We change the native vlan first on Fa0/16 of SW1 and then on Fa0/13 of SW2 to complete the configuration.



## TUN MIN OO {BE-IT} Routing & Switching 200-120

You can verify the configuration using the good old show interface trunk command on SW1 and SW2. The output on SW1 looks something like:

```
SW1#show interface trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/13	on	isl	trunking	1
Fa0/16	on	802.1q	trunking	10
Fa0/19	auto	n-isl	trunking	1

Port	Vlans allowed on trunk
Fa0/13	1-4094
Fa0/16	1,10,20,30,40,50
Fa0/19	1-4094

Port	Vlans allowed and active in management domain
Fa0/13	1
Fa0/16	1
Fa0/19	1

Port	Vlans in spanning tree forwarding state and not pruned
Fa0/13	1
Fa0/16	1
Fa0/19	1

## **7-8 Inter-VLAN Routing Configuration**

By default, a host can communicate with only those hosts that are members of the same VLAN. In order to change this default behavior and allow communication between different VLANs, you need a router or a layer 3 switch. We will learn both approaches starting with the router approach.

The router has to support ISL or 802.1Q trunking on a FastEthernet or GigabitEthernet interface in order to perform routing between different VLANs. The router's interface is divided into logical interfaces called *subinterfaces*, one for each VLAN. From a FastEthernet or GigabitEthernet interface on the router, you can set the interface to perform trunking with the **encapsulation** command:

Please note that the Cisco 2811 router named R1 supports only 802.1Q trunking. As we learned earlier in the chapter that Cisco is moving away from ISL and newer hardware like the Cisco 2800 series Integrated Services Router (ISR) does not even support ISL.

We have used subinterface number 10 which happens to be the same as the VLAN ID associated with the subinterface. It is common practice to make the subinterface number match the VLAN ID which makes the configuration more predictable and helps

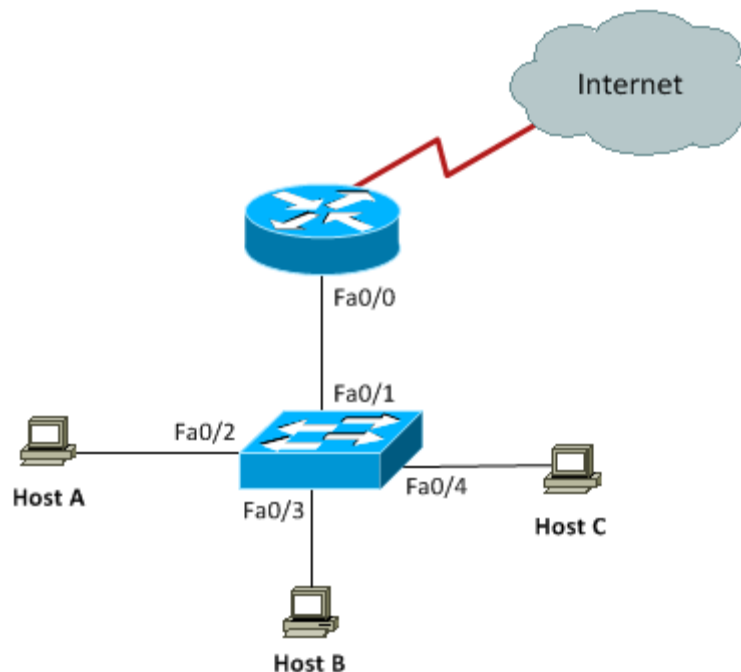
in configuration and troubleshooting. But it is just an arbitration and subinterface number and VLAN ID don't have to necessarily match. Remember that the subinterface number is only locally significant, and it does not matter which subinterface numbers are configured on the router.

Another important fact about VLANs is that each VLAN also is a separate IP subnet. Although it is not an absolute requirement to have a one-to-one mapping between VLANs and IP subnets but it really is a good idea to configure your VLANs as separate subnets, so better stick to this best practice.

In order to make sure you are fully prepared to configure inter-VLAN routing, we will go through two different configuration examples in detail.

Let's start by looking at the figure that follows and reading the router and switch configuration given for the figure.

**Figure 7-11** Inter-VLAN Routing Example1



The switch interfaces have the following roles:

On the router, you create two subinterfaces one for each VLAN matching the subinterface number with the VLAN ID associated with the subinterface. Each VLAN has its own IP subnet and the IP addresses are configured for subinterfaces. Notice that we did not configure any IP address for the physical interface on the router. This is standard router-on-a-stick configuration for inter-VLAN routing:

Having come this far in your CCNA studies, you should be able to figure out which IP subnets are being used by looking at the router configuration. You can see that we are using 192.168.10.0/24 with VLAN 10 and 192.168.20.0/24 with VLAN 20. And by looking at the switch configuration, you can see that interfaces FastEthernet0/2 and FastEthernet0/3 are in VLAN 10 and interface FastEthernet0/4 is in VLAN 20. This means that Host A and Host B are in VLAN 10 and Host C is in VLAN 20.

We are configuring the IP addresses on hosts manually or statically as below:

The hosts can have any IP address in the subnet range but I just chose the first available IP addresses after the default gateway address to make the configuration simpler and predictable. Always keep in mind that easier to read and predict configurations are always easier to maintain and troubleshoot as well from a practical standpoint.

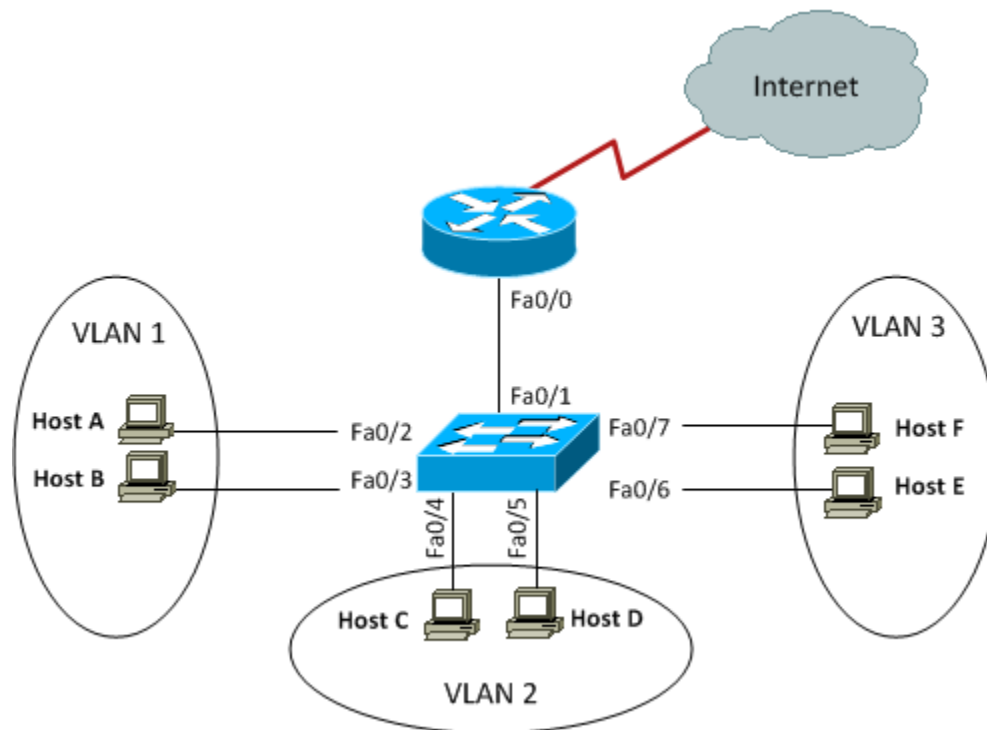
Now again using the figure as reference, let's go through the commands necessary to configure switch interface Fa0/1 to establish a link with the router and provide inter-

VLAN communication using IEEE 802.1q encapsulation. Please note that I have used a Cisco 3560 switch here and the commands can vary slightly depending on what switch model you are working with.

As you can see, our Cisco 3560 switch supports both IEEE 802.1Q and ISL encapsulation in addition to **negotiate** mode that allows encapsulation to be negotiated through dynamic Trunking Protocol (DTP). We specified 802.1Q as the trunking protocol in order to successfully perform trunking with the router. Also keep in mind that when we create a trunk link like the one we just created, all VLANs 1 to 4094 are allowed to pass data by default. However, it is possible to allow only a subset of the range of VLANs while blocking others.

Let's move on to our second and final configuration example for inter-VLAN routing involving a somewhat more complex scenario as shown in the figure below:

**Figure 7-12** Inter-VLAN Routing Example 2



This figure shows three VLANs 1, 2, and 3 with two hosts in each of them. The router is connected to the switch using subinterfaces on port Fa0/1 on the switch. The switch port connecting to the router is a trunk port. The switch ports connecting to the clients are all access ports, not trunk ports. The configuration of the switch would look something like this:

Before we configure the router, we need to know the IP subnets assigned to VLANs:

The configuration of the router would then look something like this:

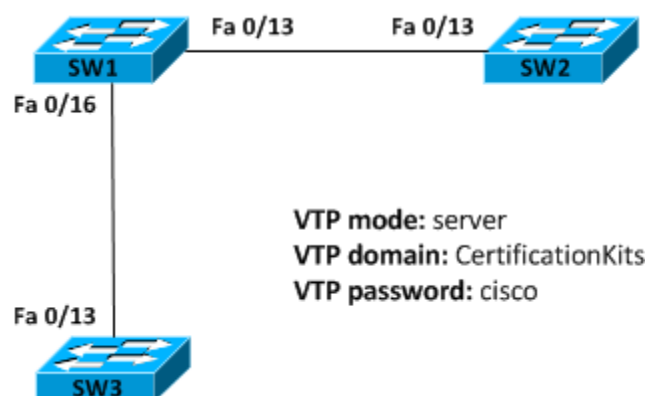


The hosts in each VLAN would be assigned an IP address from the IP subnets associated with the VLAN, and the default gateway would be the IP address assigned to the router's subinterface in that VLAN.

## VTP Configuration

In this section, we will configure VLAN Trunking Protocol (VTP) for the switched network shown in the diagram:

**Figure 7-13** VTP Configuration Example



Cisco switches are configured to be in VTP server mode by default. The first step in configuring VTP would be to set the VTP domain name you want to use. VTP domain name can be any string of characters which must be configured on all switches that are to exchange VLAN information over VTP with each other.

When you create the VTP domain, there are quite a few options you can set including the domain name, password, mode, and pruning. You can set all of these options using the **vtp** command in global configuration mode. In the following example, we will set

switch SW1 to VTP**server** mode, the VTP domain to **CertificationKits**, and the VTP password to **cisco**:

We are done with configuring various VTP options but we have to find a way to verify that configuration. There are two very useful commands to verify VTP configuration and they are **show vtp status** and **show vtp password**:

The preceding output shows that VTP mode, domain name, and password have been successfully configured. You may recall that all switches are in VTP server mode by default, and you actually have to be in VTP server mode if you want to change any VLAN information on the switch.

Let's now go to switches SW2 and SW3 and set them into the **CertificationKits** VTP domain. It is very important to keep in mind that the VTP domain name is case sensitive.

You can repeat the same configuration on SW3 to complete the configuration on all three switches. Now that all our switches are set to the same VTP domain and password, it's time to test if our VTP configuration achieves what it is supposed to achieve. You may recall that the primary goal of VTP is to be able to create VLANs only on the VTP server and let that VLAN information propagate to VTP clients through VTP advertisements. We created a few VLANs on SW1 earlier and they should be advertised to the VTP client switches SW2 and SW3 if VTP is working as expected. This can easily be verified by using the good old **show vlan brief** command on switches SW2 and SW3:

```
SW2#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/3 Fa0/4, Fa0/5, Fa0/6 Fa0/7, Fa0/8, Fa0/9 Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15 Fa0/22, Fa0/23, Fa0/24 Gi0/1, Gi0/2
10	Engineering	active	
20	Finance	active	
30	Management	active	
40	Marketing	active	
50	Sales	active	
1002	<u>fddi</u> -default	act/ <u>unsup</u>	
1003	<u>token-ring</u> -default	act/ <u>unsup</u>	
1004	<u>fddinet</u> -default	act/ <u>unsup</u>	
1005	<u>trnet</u> -default	act/ <u>unsup</u>	

As you can see five new VLANs are present on SW2 though we never did any VLAN configuration on SW2. These VLANs have been learned from the VTP server SW1 through VTP advertisements. You may have noticed in the above output that though SW2 has learnt new VLANs through VTP, no switch ports are yet assigned to these new VLANs. Keep it very clear in your mind that VTP only advertises VLAN information;

it cannot advertise VLAN port assignments. Individual switch ports must be manually assigned to desired VLANs on all switches.

## VTP Pruning

VLANs are an efficient way to preserve bandwidth by localizing broadcasts, multicasts, and unicast frames. VLAN Trunking Protocols serves the basic purpose of making VLAN management centralized and more efficient. But VTP has a small nifty feature that gives us a way to preserve bandwidth even further within a VLAN. This feature is called *pruning*. VTP pruning enabled switches send broadcasts to only those trunk links that actually must have the information. Let's explain it a bit: If SW1 does not have any ports assigned to VLAN 2 and there is a broadcast generated in VLAN 2, that broadcast would not traverse the trunk link from a connected switch to SW1. In other words, other switches connected to SW1 would not send any broadcasts generated in a specific VLAN to SW1 if SW1 has no port assigned to that VLAN, if VTP pruning is enabled.

When we enable pruning on a VTP server, you effectively enable it for the entire VTP domain. By default, only VLANs 2 through 1001 are pruning eligible, but VLAN 1 cannot be pruned because it is the default administrative VLAN. VTP pruning is disabled by default but it is a good idea to enable it to save some bandwidth. And you know what, the configuration is surprisingly simple:

The command **show vtp status** can yet be used to find VTP pruning state currently configured:

We can do similar verification on SW2:

Now, if we issue the **show interface trunk** command on SW2, we discover some interesting facts supporting our understanding of VTP pruning.

Because there are no switch ports assigned to any of the VLANs 10, 20, 30, 40 and 50 on SW2, all these VLANs have been pruned as shown in grayed output above. VLAN1 cannot be pruned being the administrative VLAN and the same is reflected in the output.

## **7-9 VTP Troubleshooting**

In a perfect world you could come up with a configuration for your switches, apply the configuration and expect it to work as you think it should. But in the real world, configurations may not work as expected more often than you believe. Thus let's review some VTP troubleshooting techniques. You will also find Cisco is starting to put more emphasis on troubleshooting on the CCNA exam.

I would give you a piece of advice here regarding troubleshooting in general. This is a mistake novice network engineers often make while troubleshooting. They use **show running-config** command to examine the configuration and try to find the fault. This is not an efficient troubleshooting technique. Effective troubleshooting involves a deep understanding of the technologies involved and thorough familiarity with troubleshooting tools available in Cisco IOS. If you are wondering which troubleshooting tools I am talking about, they are none other more relevant than the **show** and **debug** commands. For connectivity testing you can use the **ping** and **traceroute** commands in their basic and extended forms. **show** and **debug** commands provide a lot of useful information which can be used to quickly reach the source of problem by experienced troubleshooters. As you start to learn troubleshooting techniques, try to become familiar with as many of these troubleshooting commands as you can. The troubleshooting skills you acquire at this stage as you are preparing for your CCNA certification will be an asset as you possibly pursue more advanced Cisco certifications and in the real world.

Let's assume we completed out VTP configuration making SW1 a VTP server while SW2 and SW3 are VTP clients. We configured a few VLANs on SW1 and went to SW2 to verify it has learned VLAN information from SW1 by issuing **show vlan brief** command:



```
SW2#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/3 Fa0/4, Fa0/5, Fa0/6 Fa0/7, Fa0/8, Fa0/9 Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15 Fa0/22, Fa0/23, Fa0/24 Gi0/1, Gi0/2
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

It is disturbing to see that SW2 has not learned any of the VLANs we created on the VTP server. There is something wrong with our configuration which we need to troubleshoot. We are not going to do a show running-config here but rather use the show and debug commands. In fact, you should try the **show** commands first and **debug** commands are to be used as a last resort. Most problems can be isolated using show commands alone.

A good starting point is to run show vtp status and show vtp password on SW2 and check for the VTP domain name, password and mode.

```
SW2#show vtp status
VTP Version                : running VTP1 (VTP2 capable)
Configuration Revision      : 9
Maximum VLANs supported locally : 1005
Number of existing VLANs    : 10
VTP Operating Mode         : Client
VTP Domain Name            : CertificationKits
VTP Pruning Mode           : Disabled
VTP V2 Mode                : Disabled
VTP Traps Generation       : Disabled
MD5 digest                 : 0xA6 0xA3 0xBA 0x79 0xFE 0x57 0x0A
                           : 0xDC
Configuration last modified by 10.10.30.1 at 3-1-93 05:19:23
SW2#show vtp password
VTP Password: Cisco
SW2#
```

You may notice that the VTP password is set as *Cisco* rather than *cisco* which seems to be the source of problem. We can fix this:

After correcting the VTP password SW2 would learn VLAN information from the VTP server which can be verified by running **show vlan brief** command.

In brief, most VTP synchronization problems are caused by a misconfiguration of domain name, password, mode, or version and can be diagnosed by **show vtp status** and **show vtp password** commands on all switches in the VTP domain.

Also, keep in mind that a mismatched domain name has another unwanted side effect that Dynamic Trunking Protocol (DTP) is not able to negotiate trunking. If you ever find yourself in a situation where trunking is not successfully negotiated while the configuration seems correct, do check that the VTP domain name matches on the two switches.

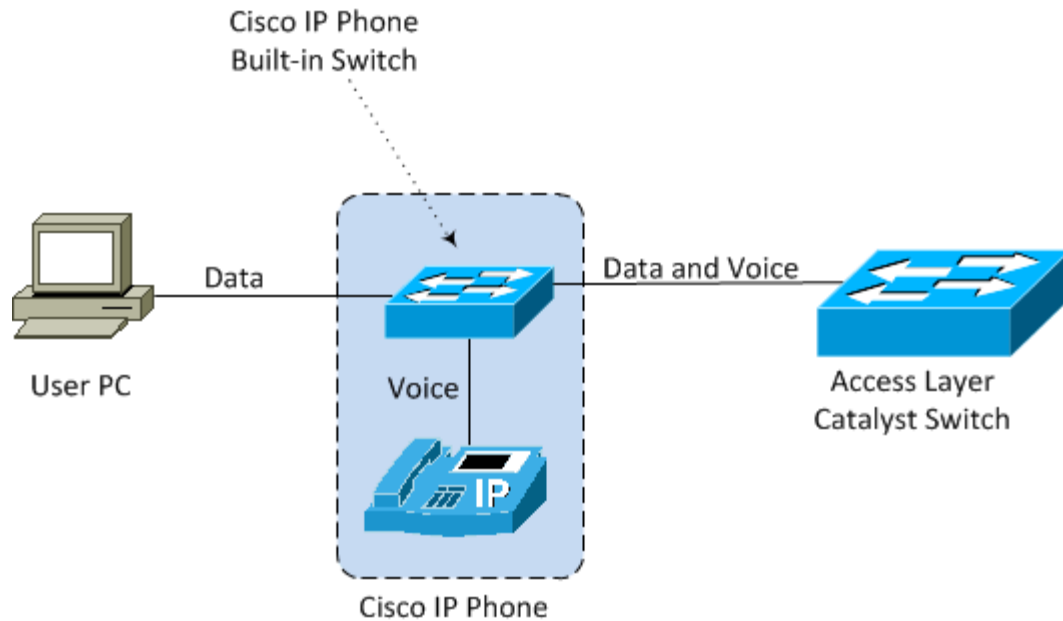
## **7-10 Voice VLAN Configuration**

We briefly covered voice access ports earlier in the chapter also mentioning voice VLANs. It is time now to dig a bit deeper into voice VLANs and do a little configuration as well.

The voice VLAN is an ingenious feature that enables access ports to carry voice traffic from an IP phone. Cisco IP phones connect to the IP network using Ethernet to send Voice over IP (VoIP) packets. The Voice over IP framework is made up of several components including IP phones, call managers, and voice gateways. A detailed coverage of these components is beyond the scope of this [BOOK](#) and your Cisco Certified Network Associate (CCNA) exam. The VoIP communication takes place over the same shared network infrastructure made up of switches and routers which is used for data communication.

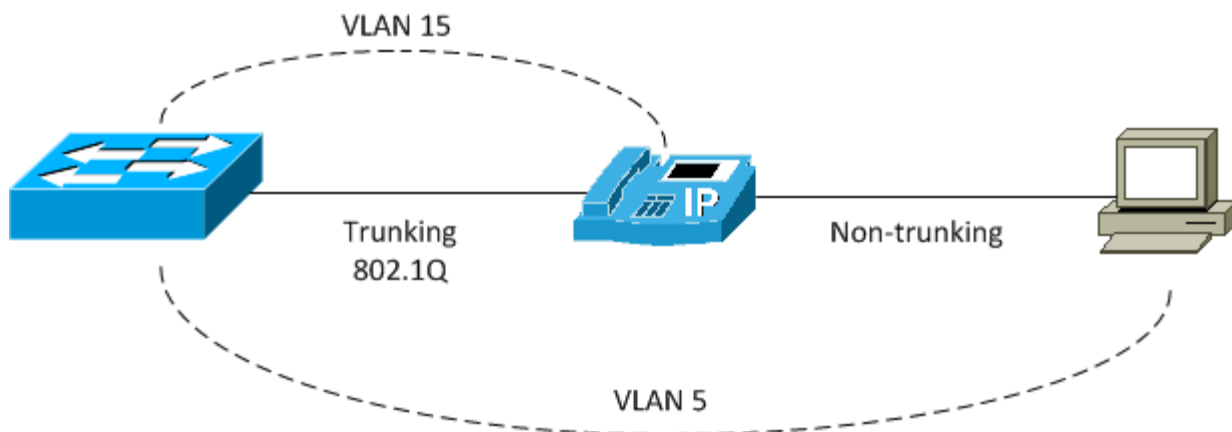
Each desk or cubicle in a modern enterprise is likely to have both an IP phone and a PC on it. One way of connecting the IP phone to switch may have been to use a separate Ethernet cable and a separate switch port. But Cisco came up with the idea of including a small LAN switch built inside each Cisco IP phone. This small switch allows one cable to run from the LAN switch to the desk to connect to the switch built into the IP phone. Then the PC can connect to the switch inside the IP phone over a short straight-through Ethernet cable from the PC to the bottom of the IP phone. If you have access to a Cisco IP Phone, turn it upside down and you would find two Ethernet ports at its bottom. One port is to be connected to the LAN switch, the second port is to be connected to the PC and the third port is internal which connects to the IP phone circuitry inside. This is the simple three port switch built into all Cisco IP phones. In this way, a Cisco IP phone provides a data connection for a user's PC, in addition to its own voice data stream. Please see figure below for a graphical representation of the concept just described.

**Figure 7-14** Built-in Switch of the Cisco IP Phone



As you can see in the diagram, the link between the phone and switch should use 802.1Q trunking, and the phone and PC should be in different VLANs and hence in different IP subnets. This design is per Cisco recommended guidelines and has several advantages. First, by placing IP phones in one VLAN, and the PCs connected to phones in a different VLAN, you can more easily manage the IP address space, apply Quality of Service (QoS), and provide better security by isolating the data and voice traffic.

**Figure 7-15** How to Connect an IP Phone and PC to LAN Switch



On a relatively quiet, underutilized network, a switch can generally forward frames as soon as they are received. However, if a network is congested, packets cannot always be delivered in a timely manner. Different types of applications have different requirements for how their data should be sent end to end. For example, it might be acceptable to wait a short time for a Web page to be displayed after a user has requested it. Also, an FTP download may continue at a variable rate without issues as user can use the file once it is fully downloaded. But it is probably not tolerable to face the same delays in receiving packets that belong to a streaming video presentation or a telephone call. Video streaming is very popular these days and typically multicast traffic over UDP as the transport protocol is used to transmit the video stream from a server to several clients. Any loss or delay in packet delivery would ruin the purpose of these applications due to their real-time or interactive nature.

Traditionally network congestion has been handled by increasing link bandwidths and enhancing switching hardware performance. This approach is not cost effective or efficient and it does nothing to address how one type of traffic can be preferred over another. Quality of Service (QoS) can be used to protect and prioritize time-critical traffic like voice and video. Keep in mind that the most important aspect of transporting voice traffic across a switched network is maintaining the proper Quality of Service level. Voice packets must be delivered in the most timely manner possible, with minimum jitter, loss, and delay.

As a matter of fact, layer 2 frames have no means to indicate the priority or importance of their contents for the purpose of prioritization or QoS. One frame looks just as important as any other frame. However, when frames are carried from switch to switch, an opportunity for classification occurs. We understand that a trunk is used to carry frames from multiple VLANs between switches. The trunk does this by encapsulating the frames and adding a tag indicating the source VLAN number. The encapsulation also includes a field that can mark the class of service (CoS) of each frame. This marking can be used at switch boundaries to make QoS decision and prioritize traffic according to importance. Cisco switches typically perform QoS implementation or traffic prioritization in hardware and the actual mechanisms may vary from platform to platform.

The LAN used for voice traffic from the IP phone is called the voice VLAN and the VLAN used for data is called the data or access VLAN. For the LAN switch to forward traffic correctly, it needs to know the VLAN ID of the voice VLAN as well as the data VLAN. The data or access VLAN is configured just as a regular access VLAN is configured using the **switchport access vlan** *vlan-id* command. The voice VLAN is configured using the **switchport voice vlan** *vlan-id* interface configuration mode command. Referring to the diagram, the switch would need both the **switchport access vlan 5** and **switchport voice vlan 15** commands in interface configuration mode.

## Summary

This chapter introduced to you a number of enhanced switching technologies and described how you can configure them on Cisco switches. We started with talking about virtual LANs (VLANs) and how they break up broadcast domains in a switched network and provide traffic isolation at layer 2. This fact is very important because layer 2 switches without VLANs only break up collision domains and your switched network is one large broadcast domain. We learned what access links are and also went over how trunked VLANs work across a Fast Ethernet or Gigabit Ethernet link.

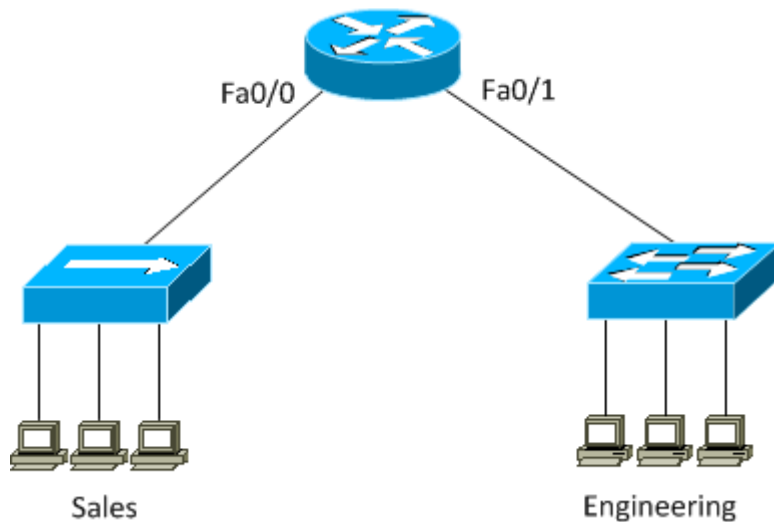
Trunking is an important and critical technology to understand as most of the enhanced switching technologies described in this chapter involve trunking one way or the other. We went into great detail describing VLAN Trunking Protocol (VTP) and learned how it sends VLAN information to all switches in the network over trunked links. We also learned how to configure and troubleshoot VTP in case things don't work as you expect them to work.

Finally we covered Voice VLANs which can be used to allow IP phones to run along with regular desktop or laptop computers over your access switch ports. We finished off

the chapter with detailed configuration and troubleshooting examples for almost all technologies covered in the chapter. **Questions**

Read the questions carefully and try to answer as many questions correctly as you can. Answers to these questions are provided on the next page.

1. Based on the exhibit shown for the local area network of an office comprising two departments, which of the following are correct ? (Choose two)



- A. There are six collision domains in the network
- B. There are two broadcast domains in the network
- C. There are four broadcast domains in the network
- D. There are six broadcast domains in the network
- E. There are five collision domains in the network

2. An Ethernet switch receives a unicast frame with a destination MAC address that is listed in the MAC address table. What will the switch do with the frame?

- A. The switch will forward the frame to a specific port
- B. The switch will forward the frame to all ports except the port on which it was received
- C. The switch will send a copy of the frame out the same port on which it was received
- D. The switch will not forward the frame at all

E. The switch will add the destination MAC address in the frame to the MAC address table

F. None of the above

3. A switch port is configured as a VLAN trunk. Which of the following trunk modes are valid ? (Select all that apply.)

A. Blocking

B. Dynamic auto

C. Dynamic desirable

D. On

E. Transparent

F. All of the above

4. Which of the following frame encapsulation methods can be configured on Cisco switch trunks? (Select two.)

A. 802.1Q

B. VTP

C. CDP

D. Auto

E. Desirable

F. ISL

5. You need to configure two switches to exchange VLAN information. Which protocol provides the functionality of sharing VLAN information between these two switches?

A. STP

B. RSTP

C. 802.1Q

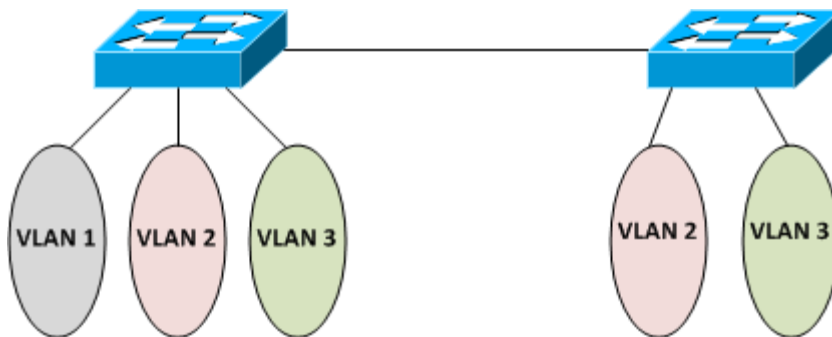
D. ISL

E. VTP

F. None of the above



**Q6.** Which of the following statements are true regarding how VLANs are used to segment a network? (Select three)



- A. VLANs increase the size of collision domains.
- B. VLANs increase the size of broadcast domain while decreasing the number of collision domains.
- C. VLANs increase the number of broadcast domains while decreasing their size.
- D. VLANs allow logical grouping of users by function.
- E. VLANs can enhance network security.
- F. VLANs simplify switch administration.

**7.** Two switches have been configured with static VLANs as shown in the figure. But VLAN 2 on switch A has no connectivity with VLAN 3 on switch B. How should the network administrator solve the problem?

- A. Configure interconnected ports on switch A and switch B in access mode.
- B. Connect the two switches using a straight-through cable.
- C. Configure VLAN 1 with IP addresses on both switches.
- D. Add a layer 3 device to provide connectivity between VLAN 2 and VLAN 3.
- D. Ensure that VTP passwords match on both switches.

**8.** Which of the following steps are basic requirements in order to add a new VLAN to a switched network?

- A. Create the VLAN.
- B. Name the VLAN.

- C. Configure an IP address for the VLAN.
- D. Add the desired switch ports to the new VLAN.
- E. Add the VLAN to the VTP domain.

9. You connect a new PC to a free port on a switch, but you find that the PC cannot access any of the resources on the LAN. No other PC connected to the switch has connectivity issues. What is the *most likely* cause of this problem?

- A. The MAC address is not configured correctly on the host.
- B. An STP instance is not running for the new host.
- C. The switch does not have the MAC address of the new host hard coded in the MAC address table.
- D. The switch port host is connected to is assigned to the incorrect VLAN.
- E. The router has not learned the route to the new host.

10. Please study the exhibit carefully. The switch has twenty four Fast Ethernet ports and two Gigabit Ethernet ports. But why are some ports missing from the list of ports assigned to the default VLAN?

```
SW1#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7 Fa0/8, Fa0/9, Fa0/10 Fa0/12, Fa0/13, Fa0/14 Fa0/15, Fa0/17, Fa0/18 Fa0/19, Fa0/20, Fa0/21 Fa0/22, Fa0/23, Fa0/24 Gi0/1, Gi0/2
100	VLAN0100	active	Fa0/1, Fa0/11
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

- A. The missing ports are administratively shut down.
- B. The missing ports are not actively participating in STP
- C. The missing ports are assigned to VLAN 100.

- D. The missing ports are configured as trunk ports.
- E. The missing ports have a speed or duplex mismatch with neighboring ports.
- F. None of the above.

### Answers

- 1. B, E
- 2. A
- 3. B, C, D
- 4. A, F
- 5. E
- 6. C, D, E
- 7. C
- 8. A, D
- 9. D
- 10. C

## **Chapter 8 – Network Security**

Vint Cerf who is recognized as one of the fathers of the Internet, once said, “The wonderful thing about the Internet is that you’re connected to everyone else. The terrible thing about the Internet is that you’re connected to everyone else.” There has been an explosion in the size and scope of the Internet and today anyone with a computer can connect to almost anyone else. Most companies too are now permanently connected to the Internet, a network through which others could also attempt to illegally access their networks. Network security has become one of the hottest topics in networking and the trend is likely to continue in the near future and why Cisco has developed the CCNA Security certification specialty.

- 8-1 Network Security
- 8-2 Cisco Firewalls
- 8-3 Layer 2 Security
- 8-4 AAA Security Services
- 8-5 Secure Device Management
- 8-6 Secure Communications

## **8-1 Network Security**

### **How to Approach Network Security**

While the Internet and networks are growing rapidly, they are also becoming more complex and mission critical. This brings new challenges to the folks who run and manage today's networks. There has been an integration of network infrastructure that now supports voice, video, and data but at the same time new security concerns are also introduced.

As a matter of fact, no computer system in the world can be completely secure no matter how good the security measures are. Probably the only way to fully secure a computer is to isolate it completely, restricting all physical and virtual access to it. Such a system would not be connected to any network and would probably be stored in a secured vault somewhere with no physical access. Though this computer system would be completely secure, it would also be completely useless. Usefulness of computers stems from the ability to connect to them and use the resources offered by them. So, the goal of network security is to provide continued access to those resources and, at the same time, preventing any un-authorized or malicious activity from taking place.

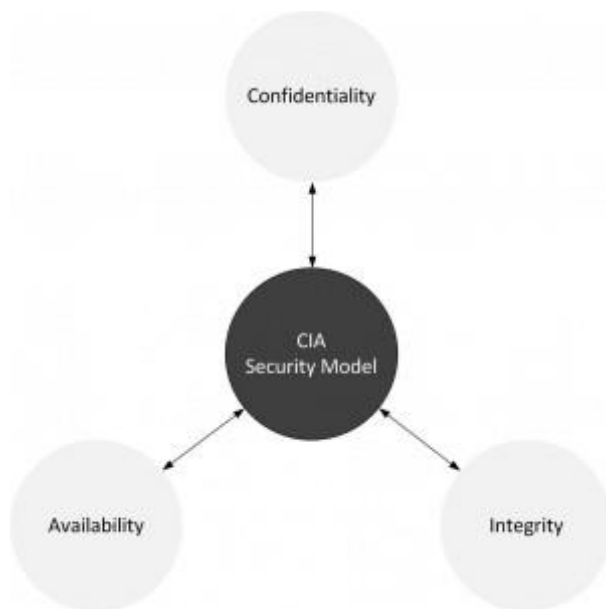
Cisco IOS software running on Cisco routers has several built-in security tools that can be used as part of a good overall security strategy. Probably the most important security tool in Cisco IOS software are access control lists (ACL). ACLs can be used to define rules to prevent some packets from flowing through the network. In this chapter, you will learn how you can protect the posterior of your network by deterring the most common threats with features available in Cisco IOS itself.

Cisco also produces an array of specialized security appliances such as the Adaptive Security Appliance (ASA) that companies can use for securing their networks.

### **The CIA Model**

A security model is a framework that provides guiding principles to make systems secure also meeting industry best practices and regulations. A widely applicable model of network security is the confidentiality, integrity, and availability (CIA) triad. CIA is more like a set of three guiding principles that can be used to secure systems. A breach of any of these three principles can have security consequences.

**Figure 8-5** CIA Security Model



## Confidentiality

*Confidentiality* means preventing sensitive information from being seen by anyone who is not authorized to see it. It is the capability to ensure that the required level of secrecy is enforced and information is concealed from unauthorized users. Information is a very valuable asset and keeping sensitive information secure is critical for enterprises. That's why confidentiality is the aspect of security that comes under attack most often by those who want to steal information for their own interests. Encryption is a common technique used to ensure confidentiality of data transferred from one computer to another. For example, when a user is performing an online banking transaction, sensitive information such as account statements, credit card numbers, and passwords must remain protected. Encryption techniques ensure that information is not seen as it is being sent back and forth between the user's computer and the online bank.

## Integrity

*Integrity* prevents any unauthorized modification of data to make sure information stays accurate. If your data has integrity, you can be sure that it is the actual unchanged representation of the original information and hence can be trusted. A common type of security attack that compromises the integrity of data is the man-in-the-middle attack. In this kind of attack, the attacker intercepts data as it is in transit and makes changes to it without letting the two communicating entities realize that.

## Availability

*Availability* prevents the loss of access to information and resources and ensure that they are ready for use when they are needed. It is a must to make sure that information is readily available at all times so that requests by authorized users could be fulfilled whenever they come. Denial of service (DoS) is one of several types of security attacks that attempts to prevent legitimate access to information and resources hence compromising the availability of affected systems.

**Table 8-1** CIA Model

Goal	Defined	Example	Methodology
Availability	Keeping your network services up and running	DoS Attacks	Auto patch updatesRate limiting
Integrity	Prevent data modification	Man in the Middle Attack	Hashing
Confidentiality	Secure Data from eavesdropping	Packet capture and replaying	Encryption

## The Secured Enterprise Network

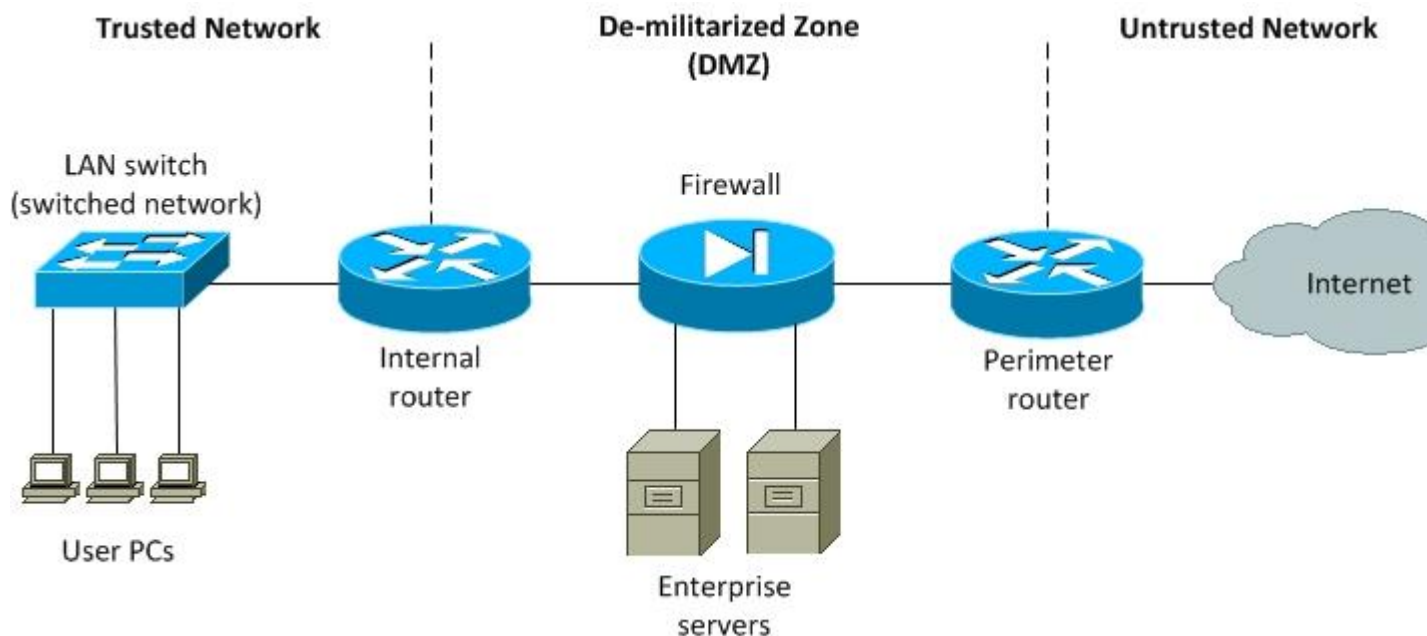
In a medium to large enterprise, the typical secured network is built around a recipe of a perimeter router, a firewall device, and an internal router.

**Perimeter Router** The *perimeter router* is the border crossing or the demarcation point between enterprise network resources and the public network, such as the Internet. Therefore, traffic originating from the outside destined for the trusted network or the DMZ must transit through the perimeter router. This router should provide basic security and traffic filtering for both the DMZ and the trusted network.

**Firewall** The *firewall* can be a router running the Cisco IOS firewall feature set or a specialized device like the Cisco Adaptive Security Appliance (ASA). The firewall can be configured to provide sophisticated controls over traffic flowing between the trusted network, DMZ, and the untrusted network.

**Internal Router** The *internal router* provides additional security by providing a point where you can apply further controls to traffic going to or coming from various parts of the trusted network.

**Figure 8-1** Secured Enterprise Network





You should do a detailed examination of Figure 8-1 and identify clearly the three distinguishable parts of the network: *trusted network*, *untrusted network*, and the *demilitarized zone (DMZ)*.

**Trusted Network** The trusted network is the internal enterprise network or the corporate local area network (LAN).

**Untrusted Network** The untrusted network refers to the universe beyond the perimeter router. Typically, the Internet is the untrusted network and is considered highly hostile.

**Demilitarized Zone (DMZ)** The term DMZ, like many other network security terms, was borrowed from military terminology. In military terms, a demilitarize zone (DMZ) is an area, usually the frontier or boundary between two or more military powers, where military activity is not permitted, usually by peace treaty or other similar agreement. In computer networking, the DMZ likewise provides a buffer zone that separates an internal trusted network from the untrusted hostile territory of the Internet. DMZ is not as secured as the internal network, but because it is behind a firewall, neither is it as non-secure as the Internet. Typically, DMZ hosts services to which access is required from the untrusted network. This includes Web, DNS, email, and other corporate servers that have to be reachable from the Internet.

## Classes of Attackers

In the context of this chapter, an attacker refers to someone who attempts to gain unauthorized access to a network or computer system. It is useful to identify different types of attackers and understand their motives in order to be able to characterize attacks and track down such individuals. There are a variety of groups into which attackers are classified and sometimes conflicting views are held by members of the networking community about the definitions of these classifications. Here, I would mention three broad categories:

**Hackers** Hackers are those individuals who break into computer systems and networks to learn about them, or just to prove their prowess. Some hackers usually mean no harm and do not seek financial gain.

**Crackers**      Crackers are criminal hackers who intend to harm information systems. Crackers usually work for financial gain and are also known as black hat hackers.

**Script Kiddies**      Script kiddies think of themselves as hackers but do not have the needed knowledge and skills. They cannot write their own code; instead, they run scripts written by others to attack systems and networks. As a matter of fact, very sophisticated software tools have become freely available on the Internet which allow novices to execute attacks with point-and-click ease. Today, a very large percentage of wannabe hackers fall in this category.

### **Vulnerabilities, Threats, and Exploits**

Security attacks vary considerably in their sophistication and ability to do damage. As you would learn more about protocols that run today's networks, you would realize that most security threats are a result of some weakness or inadequacy in the design of the underlying protocol itself. When the Internet was formed, it linked various government entities and universities to one another with the sole purpose of facilitating learning and research. The original architects of the Internet had never anticipated the kind of widespread adoption the Internet has achieved today. As a result, in the early days of networking, security was not designed into network protocol specifications. For this reason most implementations of TCP/IP are inherently insecure. That is a big reason why security is such a burning issue today and in the absence of built-in security mechanisms, we have to rely on additional security measures to make communications secure.

**Vulnerability**      A vulnerability is a weakness in a system or its design that can be exploited by a *threat*. Vulnerabilities are found in operating systems, applications, and even in network protocols themselves.

**Threat**      A threat is an external danger to the system having a vulnerability.

**Exploit**      An exploit is said to exist when computer code is actually developed to take advantage of a vulnerability. Suppose that a vulnerability exists in a piece of

software but nobody has yet developed computer code to abuse it. Because there is no exploit, there is no real problem yet though the vulnerability exists theoretically.

### Classes of Attacks

The three major types of network attacks, each having its own specific goal, are as follows:

#### Reconnaissance Attacks

Reconnaissance literally means the military observation of a region to locate an enemy or to establish strategic features of the region. A reconnaissance attack is not meant to inflict immediate damage to a system or network but only to gather information about the network to prepare for a later attack. It is used to map out the network and discover which IP address ranges are used, which systems are running, and which services or applications reside on those systems. The attacker has to be able to *reach* a system or network to some extent to perform reconnaissance, but normally no damage is caused at that time. The more common reconnaissance attacks include ping sweeps, port scans, and DNS queries. Here are a few examples of reconnaissance attacks:

**Information Lookup** A network intruder can use tools such as the **nslookup** and **whois** in order to determine the IP address space assigned to an organization. Finding a target IP address is one of the first steps in reconnaissance. Once an IP address range is known, an intruder can look for hosts that are alive using ping sweeps. Finally, port scanning can be used to find out which services or applications are running on those live hosts.

**Ping Sweeps** A ping sweep is a scanning technique used in the reconnaissance phase of the attack, to determine live hosts or computers in a network. A ping sweep sends ICMP echo requests to multiple hosts one after the other. If a certain address is live, it will return an ICMP echo reply confirming its existence.

**Port Scans** Port scanning is a method used to enumerate what services and applications are running on a system. An intruder sends random requests on different

ports and if the host responds to the request, the intruder gets confirmation that the port is active and the associated service or application is listening. The attacker can then proceed to exploit any vulnerabilities by targeting active services. A port scanner is a piece of software designed to search a network host for open ports. Ping sweeps and port scans are two primary reconnaissance techniques used to discover hosts and services that can be exploited.

**Packet Sniffers** A packet sniffer is a software program that uses a wired or wireless network interface card (NIC) in promiscuous mode to capture all network packets that are sent across a particular collision domain. Promiscuous mode is a mode in which the network interface card sends all packets received on the network to an application for processing. You may recall that, a network interface card would normally send only frames addressed to the MAC address of the card or broadcast / multicast frames to an application while all other frames are simply ignored. There are legitimate applications of network sniffers in troubleshooting and network traffic analysis. However, there are several network applications like Telnet, FTP, SMTP and HTTP that send data in clear text. A packet sniffer can capture all data these applications send including sensitive information, such as user names and passwords. Packet sniffing is essentially eavesdropping and the information gathered can be used to execute other attacks.

## **Access Attacks**

An access attack is meant to exploit a vulnerability and gain unauthorized access to a system on the network. The information gathered by reconnaissance attacks is used to execute an access attack. When unauthorized access is gained, the attacker can retrieve, modify, or destroy data as well as network resources including user access. Even worse, the attacker can plant other exploits on the compromised system that can be used later to gain access to the system or network with relative ease. Some examples of access attacks are detailed below.

**Password Cracking** Password cracking is very attractive for attackers as passwords are used to protect all kinds of information including online bank accounts. Password attacks can be accomplished using several methods, including brute forcers, Trojans, IP spoofing, and packet sniffers.

**Man-in-the-middle Attacks**      The man-in-the-middle (MITM) attack, also known as TCP hijacking, occurs when an intruder intercepts communication between two points and can even modify or control the TCP session without the knowledge of either party. TCP hijacking affects TCP based applications such as Telnet, FTP, SMTP (email), or HTTP (Web) sessions.

**Trojans**      A Trojan or Trojan horse is a malicious program that is hidden inside another useful application. Trojans are seemingly harmless programs that hide malicious programs such as a key logger that could capture all keystrokes including passwords, without the knowledge of the user. The term Trojan Horse has originated from the hollow wooden statue of a horse in which a number of Greeks are said to have concealed themselves in order to enter and conquer the ancient city of Troy.

**Key Logger**      A key logger is a tool designed to log or record every single keystroke on the target computer, in a covert manner so that the person using the keyboard is unaware that their actions are being monitored. All kind of information including sensitive information like password has to be basically typed on a computer. Key loggers can log and store all such information on the same computer which can either be retrieved manually or sent as an automated email by the key logger itself. Keyloggers can be both software and hardware based. Several financial institutions use on-screen keyboards for online access to customer accounts as a precaution against keyloggers.

**Trust Exploitation**      The goal of trust exploitation attack is to compromise a trusted host, so that it could be used to stage attacks on other hosts in a network. Typically hosts inside the network of an enterprise are protected by a firewall placed at network boundary. So it is difficult to attack these internal hosts from outside. But these hosts are sometimes made accessible to a trusted host outside the firewall for legitimate purposes. If this trust outside host is compromised, it can be used to attack the inside hosts with relative ease.

**Port Redirection**      A port redirection attack is a kind of trust exploitation attack that uses a compromised but trusted host to pass traffic through a firewall that would otherwise be blocked. Outside hosts can legitimately reach the DMZ and hosts in the

DMZ can legitimately reach both inside and outside hosts. If an attacker is able to compromise a host in the DMZ, he could install software to redirect traffic from the outside host directly to the inside host. This would result in outside host gaining illegitimate access to inside hosts without violating the rules implemented in the firewall. An example of a utility that can provide this type of access is netcat.

**Rootkits** The term rootkit is made up of the word *root* which is the traditional name of the privileged account on Unix / Linux operating systems, and the word *kit* which refers to the software components that implement the tool. When a malicious software providing unauthorized access is installed on a system, it is also important to hide the existence of such software to enable continued privileged access. A rootkit is designed to do just that: hiding the existence of certain processes or programs from normal methods of detection. An attacker can install a rootkit when they have obtained root or administrator access to the target system as a result of a direct attack.

**Viruses** A virus is a malicious software program or code that can cause damage to data or other programs on the target system.

**Worms** A worm is similar to a virus but it is capable of self-replication increasing the scope of its damage. Worms actually are viruses that can reside in the active memory of a system and can self-replicate and self-propagate from one computer to another over the network.

**Buffer Overflows** Buffers are locations in computer memory that are used to temporarily hold data and code. A buffer overflow occurs when a program attempts to store data in a buffer, but data is larger than the size of the allocated buffer.

**IP Spoofing** IP spoofing happens when an intruder attempts to disguise itself by pretending to have a source IP address of a trusted host in order to gain access to resources on a trusted network. Using an IP address of another known host or known network, the attacker attempts to send and receive traffic on the network. The attacker is then able to use network resource that are associated with that specific IP address. Once the attacker has got access with IP spoofing, he can use this access for many purposes.

### **Address Resolution Protocol (ARP) Spoofing**

ARP spoofing occurs when an attacker tries to disguise its source MAC address to impersonate a trusted host.

Address Resolution Protocol (ARP) is used to map IP addresses to MAC addresses residing on one LAN segment. When a host sends out a broadcast ARP request to find the MAC address of a particular host with known IP address, an ARP response comes from the host whose IP address matches the request. The ARP response is stored by the requesting host. An attacker can abuse this mechanism by responding as though they are the requested host.

### **Denial of Service (DoS) Attacks**

A Denial of Service (DoS) attack is designed just to cause an interruption to a system or network temporarily, denying access to legitimate users. This interruption in turn can cause loss of money and reputation by preventing customer access to online services. These attacks usually target specific services and attempt to overwhelm them by making numerous requests concurrently. If a system is not protected to react to a DoS attack, it can be easily brought down by running scripts that generate a very large number of requests. Some examples of Denial of Service (DoS) attacks are detailed below.

### **Distributed Denial of Service (DDoS)**

It is possible to greatly increase the impact of a DoS attack by launching the attack from multiple systems (botnets) against a single target. This scaled up version of DoS is referred to as a distributed DoS (DDoS) attack. Web servers are a popular target of DDoS attacks and DDoS attacks against companies like online retailers and Web portals keep making news headlines from time to time.

### **TCP SYN Attack**

Transmission Control Protocol (TCP) is a popular transport protocol used by several applications including Web based services. TCP is a connection oriented protocol that uses a three-way handshake to establish a TCP connection before application data exchange starts to take place. TCP SYN attack occurs when a host sends a large number of TCP/SYN packets to the target system. Each TCP SYN packet is handled like a connection request, causing the server to send back a TCP/SYN-ACK to acknowledge the connection request also maintaining the

state of this connection. The server now waits for the third packet in TCP handshake from the host initiating the connection. However, because it is not a legitimate host that initiated the connection, the third packet needed to complete the TCP handshake never arrives. These half-open connections exhausts the resources at the server, keeping it from responding to connection requests from legitimate users.

**Smurf Attack** A smurf attack occurs when the broadcast address of a network is used to send packets to all hosts on that network. If network devices are not configured properly they allow such packets to be forwarded till they reach the target network. In such an attack, the attacker will send a large number of IP packets with spoofed IP address of one of the legitimate hosts on the target network. As a result all hosts on the network receiving the broadcast would respond by a unicast sent to the spoofed local IP address. This would cause a large number of packets sent to that hosts essentially resulting in a DoS attack on that host.


### Security Threat Mitigation

Several vendors such as Check Point, Juniper, Palo Alto Networks, McAfee, Fortinet and last but not least, Cisco provide hardware and software solutions to mitigate security threats. Cisco offers a specialized yet versatile security product called the Adaptive Security Appliance or ASA, which I believe is one of the best products in its class. A Cisco ASA device is a standalone hardware security appliance. Depending on model, they are quite expensive too. Currently they are beyond the scope of the CCNA Routing and Switching exam.

Fortunately, Cisco Integrated Services Routers (ISRs) like 800, 1800, 2800, and 3800 series and the second generation (G2) of ISRs like 1900, 2900, and 3900 series also have many of the same features that are available on the Cisco ASA devices. These features are bundled as feature sets in the Cisco IOS Software that runs on these routers and include IOS Firewall, IPSec VPN, Intrusion Prevention System (IPS), and Content Filtering to mention a few. For small businesses and enterprise branch offices,



where customers are not willing to invest in a dedicated security appliance, Cisco IOS can provide much of the same functionality without additional cost.

Another basic but very powerful security tool available on Cisco IOS are the access control lists (ACLs). We will take the time to cover ACLs in great depth, in the relevant chapter of this [BOOK](#) , learning how to create and use them to mitigate security threats.

### Physical and Administrative Security Measures

The facility or physical location where devices are housed is in most cases the first and last barrier encountered by an intruder. Physical security prevents intruders from gaining physical access to the devices, and this means hands-on contact. Physical security is even more important than network security but is often overlooked by network administrators. Despite all the high level security measures, a compromise in physical access will almost always result in a complete compromise. Having a secured physical facility that is accessible only to authorized personnel is extremely important.

While trying to secure a network environment with technical measures, it is equally important to put physical and administrative security measures in place. Some examples of physical security measures are:

- Locks
- Biometric access systems
- Security guards
- Intruder detection systems
- Safes
- Racks
- Uninterruptible power supplies (UPS)
- Fire suppression systems
- Positive air-flow systems

Many security incidents emerge from the inside of the enterprise caused by employees either deliberately or un-knowingly. Policy and procedure driven administrative security measures can be effective against these threats. These administrative controls that help with information security are usually documented in the human resources (HR) department. Some of these measures are:

- Security awareness training
- Security policies and standards
- Change control mechanisms
- Security audits and tests
- Good hiring practices
- Background checks for employees and contractors

For example, if an organization has strict hiring practices that require drug testing and criminal background checks for all employees, the organization will likely hire fewer individuals of dubious character. With fewer people of dubious character working for the company, it is likely that there will be fewer internal security incidents. These administrative measures do not ensure that no security incidents would take place, but they are an important part of an information security program are often utilized especially by large organization.

## **8-2 Cisco Firewalls**

Firewalls are a very important component of any network security framework, and it is no surprise that Cisco offers firewall solutions in different shapes and forms:

- Cisco IOS Firewalls
- Cisco PIX 500 Series of Firewalls
- Cisco ASA 5500 series Adaptive Security Appliances
- Cisco Firewall Services Module

The following sections describe these platforms in more detail.

### **Cisco IOS Firewalls**

A Cisco IOS firewall is a specialized feature of Cisco IOS Software that runs on Cisco routers. It is a firewall product that is meant for small and medium-sized businesses as well as enterprise branch offices.

The earlier Cisco IOS firewall feature was called Context-Based Access Control (CBAC), which applied policies through **inspect** statements and configured access control lists (ACL) between interfaces. The Zone-Based Policy Firewall (ZBPFW) is the newer Cisco implementation of a router-based firewall that runs in Cisco IOS Software. It was introduced in IOS Release 12.4(6)T and takes advantage of many new features that make the configuration and implementation of a firewall easier than was available previously. The following are some of the important features of a Cisco IOS Firewall:

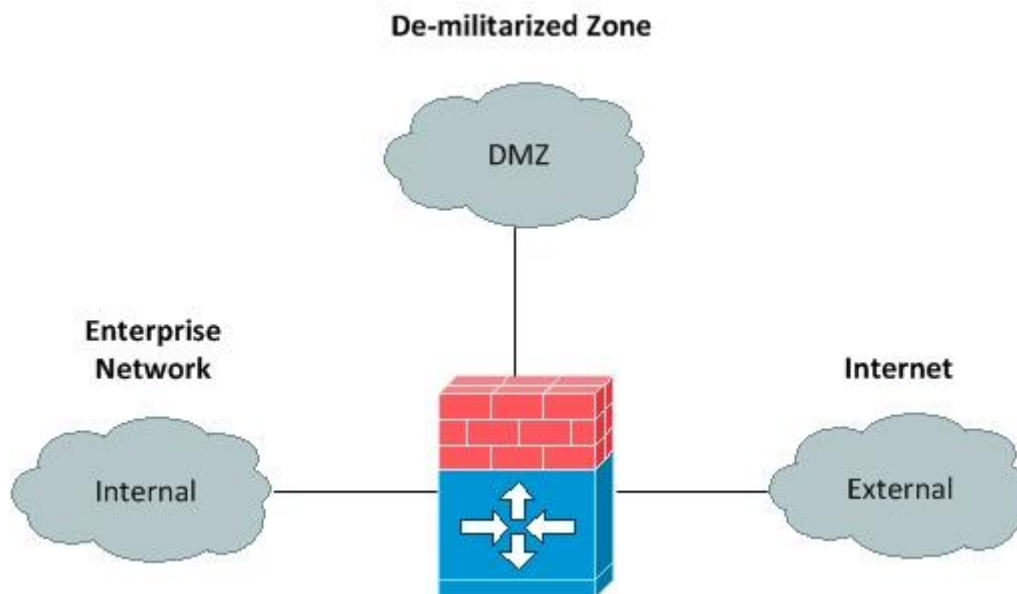
- Zone-based policy framework for easy to understand policy management
- Controlling traffic for Web, email, and other applications
- Instant messenger and peer-to-peer application filtering
- Controlling traffic for Voice over IP (VoIP) protocol
- Wireless integration
- Support for local URL whitelist and blacklist

A firewall is basically used to enforce an access policy between different security domains. With the ZBPFW feature, these different security domains are called *security zones*. With the earlier Context-Based Access Control (CBAC) feature, these security domains were simply router interfaces. So, one of the main differences between a firewall using CBAC and ZBPFW is the use of *security zones*. These zones separate the specific security areas within a network. A typical example would be a firewall that divides its universe into three main security zones:

- **Internal:** Internal or private enterprise network
- **DMZ:** Where publicly accessible servers are located
- **External:** Includes all outside destinations

Figure 8-2 describes the three primary security zones.

**Figure 8-2** Basic Zones in a Zone-Based Firewall



### Cisco PIX 500 Series Security Appliances

The Cisco PIX 500 series family of security appliances is an older series which consists of five models: the PIX 501, 506E, 515E, 525, and 535. These different models are designed to meet a range of requirements and network sizes. The Cisco PIX 500 series

security appliance provides robust policy enforcement for users and applications, secure connectivity, and multivector attack protection. These appliances provide the following integrated security and networking services:

- Firewall services with advanced application awareness
- Voice over IP (VoIP) and multimedia security
- Site-to-site and remote-access IPsec VPN connectivity
- Intelligent networking services and flexible management model

In January 2008, Cisco announced the End-of-Life for the PIX products. However, there is a large install base and Cisco will still be supporting this product until July 2013.

### **Cisco ASA 5500 Series Security Appliances**

Cisco ASA 5500 series Adaptive Security Appliances integrate firewall, Cisco Unified Communications (voice and video) security, Secure Sockets Layer (SSL) and IPsec VPN, Intrusion Prevention System (IPS), and content security services in a flexible, modular product family. The ASA 5500 series appliances provide intelligent threat defense and secure communications services that stop attacks before they affect business continuity.

The Cisco ASA 5500 series appliances are available in five models: the Cisco ASA 5505, 5510, 5520, 5540, and 5550 in order to provide a scalable security solution to meet a range of requirements and network sizes.

### **Cisco Firewall Services Module**

The Cisco Firewall Services Module (FWSM) is an integrated firewall module for high-end Cisco Catalyst 6500 switches and Cisco 7600 series routers used by large enterprises and service providers. You can install up to four FWSMs in a single switch chassis. Cisco FWSM is based on Cisco PIX firewall technology, and offers unmatched security, reliability, and performance.

### **Firewall Best Practices**

Best practice documents are a useful resource as they put together the composite effort and experiences of practitioners. Here is a generic list of best practices for your firewall security policy, which you can use as a starting point:

- Firewalls are a core security device, but you should not rely only on a firewall for security.
- Firewalls should be placed at key security boundaries.
- Your firewall policy should deny all traffic by default and services that are needed should be explicitly permitted.
- All physical access to the firewall device should be tightly controlled.
- Firewall logs should be regularly monitored accordingly to a schedule to make sure anomalies are detected.
- Proper change management procedures should be followed for firewall configuration changes, to ensure all changes are documented and no unauthorized changes take place to firewall configuration.
- A firewall primarily is a perimeter device protecting from attacks originating from the outside. It cannot protect from attacks emanating from the inside.

### **Cisco Security Appliances & Applications**

In addition to various flavors of firewalls we covered in the last few sections, Cisco also produces some other security appliances and applications to meet specific enterprise security needs.

- Cisco IronPort Security Appliance
- Cisco NAC Security Appliance
- Cisco Security Agent

### **Cisco IronPort Security Appliances**

Cisco IronPort security appliances protect enterprises against internet threats, with a focus on email and web security, which happen to be two of the main sources of endpoint threats.

The three major IronPort security appliances are:

- **IronPort C-series:** Email security appliances
- **IronPort S-series:** Web security appliance
- **IronPort M-series:** Security management appliance

### Cisco NAC Security Appliances

The purpose of Cisco Network Access Control (NAC) is to allow only authorized and compliant systems to access the network and to enforce network security policy. In this way, Cisco NAC helps maintain network stability. NAC provides four key features:


- Authentication and authorization
- Evaluation of an incoming device against network policies
- Isolating or quarantining non-compliant systems
- Remeiation of non-compliant systems

The Cisco NAC appliance condenses the four key NAC functions just described into a single appliance form and provides a turnkey solution to control network access. This solution is a natural fit for medium-scale networks that require a self-contained, ready-to-use solution. Cisco NAC appliance is especially ideal for organizations that need simplified and integrated tracking of operating system and antivirus patches and vulnerability updates. Cisco NAC appliance does not require a Cisco network to operate.

The goal of Cisco NAC appliance is to admit to the network only those hosts that are authenticated and have had their security posture examined and approved. The net result of such a thorough examination before allowing connectivity is a tremendous reduction in total cost of ownership (TCO) because only known, secure machines are allowed to connect. Therefore, laptops that have been on the road for weeks and have possibly been infected or were unable to receive current security updates cannot connect into the network and unleash a Denial of Service (DoS) attack.

Cisco NAC Appliance extends NAC to all network access methods, including access through LANs, remote-access gateways, and wireless access points. The Cisco NAC Appliance also supports posture assessment for guest users.

Cisco NAC Appliance provides the following benefits:

- It recognizes users, their devices, and their roles in the network. This occurs at the point of authentication, before malicious code can cause damage.
- It evaluates whether machines are compliant with security policies. Security policies can include specific antivirus or ANTISPYWARE  software, operating system updates, or patches. The Cisco NAC Appliance supports policies that vary by user type, device type, or operating system.
- It enforces security policies by blocking and isolating non-compliant machines. A network administrator will be advised of the non-compliance and will proceed to repair the host.

Non-compliant machines are redirected into a quarantine area, where remediation occurs at the discretion of the administrator.

### Cisco Security Agent

Cisco Security Agent is a host intrusion prevention system (HIPS) product. It is software that is installed on a server, desktop, or point-of service computing systems and provides endpoint security by its threat protection capabilities. A single management console of Cisco Security Agent can support up to 100,000 agents, so it is a highly scalable solution.

The Cisco Security Agent architecture consists of two components:

- **Management Center for Cisco Security Agents:** Management Center for Cisco Security Agent enables you to divide network hosts into groups by function and security requirements, and then configure security policies for those groups. Management Center for Cisco Security Agent can maintain a log of security violations and send alerts by email.



- **Cisco Security Agent:** The Cisco Security Agent component is installed on the host system and continuously monitors local system activity and analyzes the operations of that system. Cisco Security Agents takes proactive action to block attempted malicious activity and polls the Management Center for Cisco Security Agent at configurable intervals for policy updates. Obviously, the Management Center should also run CSA.

When an application needs access to system resources, it makes an operating system call to the kernel. Cisco Security Agent intercepts these operating system calls and compares them with the cached security policy. If the request does not violate the policy, it is passed to the kernel for execution.

However, if the request does violate the security policy, Cisco Security Agent blocks the request and takes the following actions:

- An appropriate error message is passed back to the application.
- An alert is generated and sent to the Management Center for Cisco Security Agent.

Cisco Security Agent correlates this particular operating system call with the other calls made by that application or process, and correlates these events to detect malicious activity.

### **8-3 Layer 2 Security**

Network security is only as strong as the weakest link, because a single weak point if exploited successfully would be enough for an intruder. That weak link can be the data link layer or layer 2 of the OSI reference model. We can secure the perimeter of our network protecting it from external threats but it is equally important to secure the interior of the network as several threats actually originate from the inside. Like routers, Cisco switches too have their own set of network security requirements. As a matter of fact switches may turn out to be that weak area if not properly secured. Access to switches can be a convenient entry point for attackers who want to gain access to a corporate network. With access to a switch, an attacker can launch all types of attacks from within the network. The security mechanisms that are meant to protect network perimeter would not be enough to stop these attacks simply because they originate from inside the network. For example attackers can spoof the MAC and IP addresses of critical servers to do a great deal of damage. They can even set up rogue wireless access points to provide continued access.

#### **Port Security**

You can use the port security feature on Cisco switches to restrict who can access the network by connecting to a switch port. This feature is used to limit and identify the MAC addresses of the systems that are allowed to access the port. You can configure a switch port to be secure and can also specify which MAC addresses are allowed to access the port. The secure switch port does not forward frames with source MAC addresses outside the group of defined MAC addresses for that port.

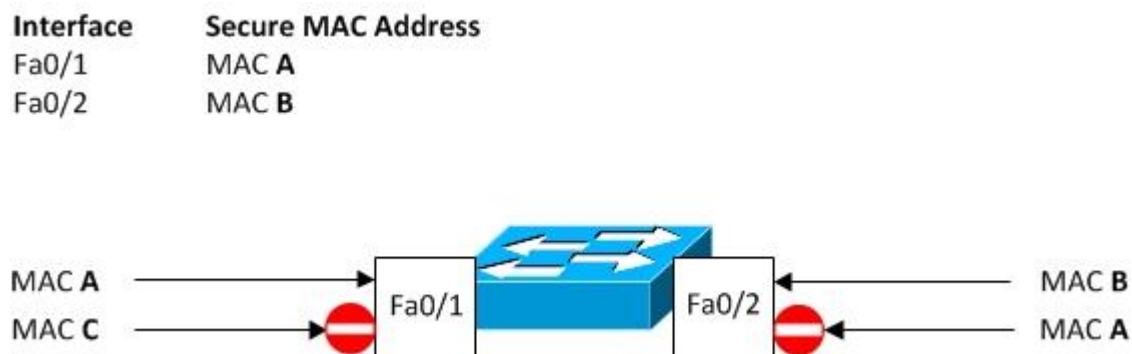
Port security allows you to manually specify MAC addresses for a port or permit the switch to dynamically learn a limited number of MAC addresses from incoming frames. By limiting the number of permitted MAC addresses on a port to just one, you can you

can make sure that just one system can connect to the port, preventing any unauthorized expansion of the network by attaching a hub or switch.

When a secure port receives a frame, the source MAC address of the frame is compared to the list of secure MAC addresses associated with the port. These secure MAC addresses are either manually configured or auto-configured or learned on the port. If the source MAC address of a frame differs from the list of secure addresses, the port either shuts down or the port drops incoming frames from unauthorized host. The default behavior of a secure port is to shut down until it is administratively enabled. The behavior of the port depends on how you configure it to respond to a security violation.

In Figure 8-3, switch port Fa0/1 will only allow those incoming frames that have source MAC address of MAC A. This port will block traffic with source MAC address of MAC C or any other frame having a source MAC address other than MAC A. Similarly, port Fa0/2 will allow traffic with source MAC address of MAC B only. This port will block all other source MAC addresses including MAC A. Despite the fact that MAC A is allowed on port Fa0/1, it is blocked on port Fa0/2 because secure (allowed) MAC addresses are specific to individual switch ports.

**Figure 8-3** Port Security



I would strongly recommend configuring the port security feature to shut down a port instead of just dropping packets from hosts with unauthorized addresses. If port security does not shut down a port, it is still possible that the port will be disabled due to too much traffic load from an attack.

Port security is a useful feature as it protects against too many MAC addresses per ports and can dictate which MAC address is allowed to connect against which port. However, if the hacker knows which MAC address is permitted on that port, he will gain access to the network by spoofing the MAC address. Port security also prevents unauthorized extension of the LAN in case a user decides to attach a hub to connect additional hosts. You have to allow only a single MAC address on the secure port to prevent this sort of extension. Also, if you are concerned about spoofed MAC addresses to bypass port security, then consider implementing IEEE 802.1X authentication mechanism.

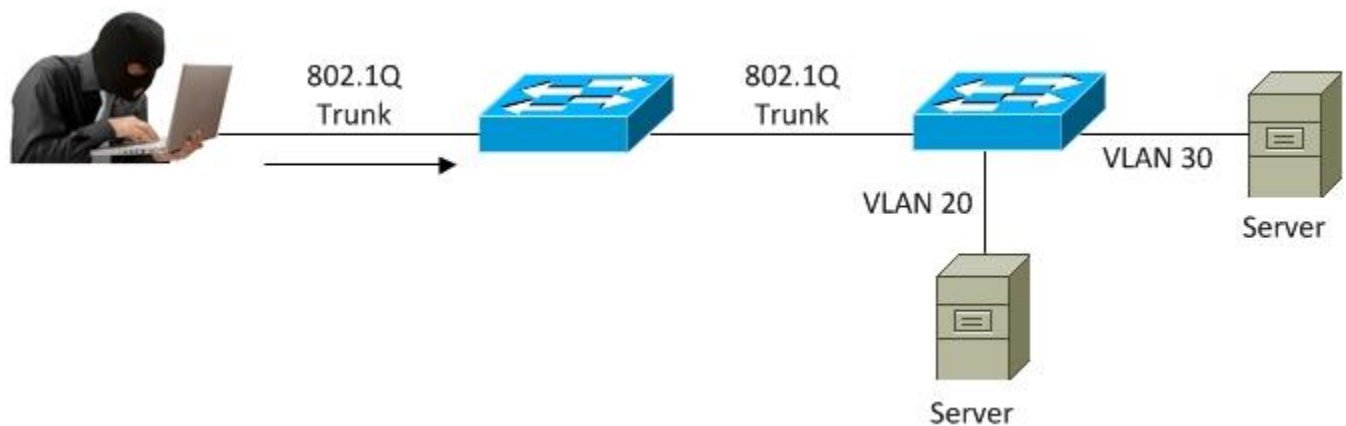
Let's see how we can configure a switch port with one specific secure MAC address. If any other device plugs into this interface not using that specific MAC address, the port will go into an err-disabled state that must be cleared by an administrator.

### VLAN Hopping

VLANs simplify network maintenance, improve performance, and provide security by isolating traffic from different VLANs. You may recall, that inter-VLAN communication is not possible without going through a router. But, a technique known as *VLAN hopping* allows traffic from one VLAN to be seen by another VLAN without first crossing a router. In some situations, attackers can even sniff data and obtain passwords and other sensitive information. The attack works by taking wrongful advantage of an incorrectly configured trunk port. As you learnt, trunk ports pass traffic from all VLANs (1 – 4094) across the same physical link, generally between switches. The data frames moving across these trunk links are encapsulated with IEEE 802.1Q or ISL to identify which VLAN a frame belongs to.

We will discuss a basic VLAN hopping attack that uses a rogue trunk link, as shown in Figure 8-4. In this attack, the attacker takes advantage of the default automatic trunking configuration found on most switches. The attacker first gets access to a vacant port on a switch and then configures a system, most likely a laptop computer, present itself as a switch. It is possible to do so if the system is fitted with an 802.1Q or ISL capable NIC, using appropriate software that usually comes with the NIC itself.

**Figure 8-4** VLAN Hopping Attack



The attacker communicates with the switch with Dynamic Trunking Protocol (DTP) messages, trying to trick the switch into thinking it is another switch that needs to trunk. If a trunk is successfully established between the attacker's system and the switch, the attacker can gain access to all the VLANs allowed on the trunk port. In order to succeed, this attack requires a switch port that supports trunking such as **desirable** or **auto**. The end result is that the attacker is a member of all the VLANs that are trunked on the switch and can *hop* on all those VLANs, sending and receiving traffic.

This sort of simple but effective VLAN hopping attack can be launched in one of two ways:

- Generating DTP messages from the attacking host to cause the switch to establish a trunk with the host. Once a trunk is established, the attacker can send

and receive traffic tagged with the target VLAN to reach any other host like a server in that VLAN, because the switch then delivers packets to the destination.

- Introducing an actual rogue switch and turning trunking on can also establish a trunk with the victim switch. The attacker can then access all the VLANs on the target switch from the rogue switch.

The best way to prevent a basic VLAN hopping attack is to turn off trunking all switch ports except the ones that specifically require trunking. All user ports should be configured with the following commands:

- **switchport mode access:** This command permanently sets the switch port to a non-trunking mode and is just good enough for all switch ports supposed to be connected to user PCs.
- **switchport nonegotiate** This command can additionally be used to disable generation of DTP messages. Though, a port configured with **switchport mode access** can never become a trunk, yet disabling DTP reduces unwanted DTP frames on the link.

On switch ports that do require trunking, DTP should be disabled using command **switchport nonegotiate** and trunking should be manually configured using command **switchport mode trunk** in interface configuration mode.

## **8-4 AAA Security Services**

AAA is an acronym that stands for authentication, authorization, and accounting:

- **Authentication:** *Who is the user?* Authentication is used to verify the identity of a user.
- **Authorization:** *What can the user do?* Authorization is used to determine what services the user can use.
- **Accounting:** *What did the user do?* Accounting performs an audit of what a user is actually doing.

AAA is a security framework that can be used to set up access control on Cisco routers, switches, firewalls, and other network appliances. AAA provides the ability to control who is allowed to access network devices and what services the user should be allowed to access. AAA services are commonly used to control telnet or console access to network devices.

AAA uses RADIUS, TACACS+, and Kerberos as authentication protocols to administer its security functions. A network device such as a router requiring AAA services establishes a connection to the security server using one of these three protocols. The security server is a Windows or Linux host external to the network device, and contains a database containing user names and passwords among other parameters. AAA on a Cisco network device can also be configured to use a local database of user names and passwords. AAA is enabled using the global configuration command **aaa new-model**.

In addition to AAA, several other simpler and less elaborate measures are available to achieve network access control, including the following:

- Local username authentication
- Enable password authentication
- Line password authentication

## **8-5 Secure Device Management**

It is important ensure the security of management traffic between a network device and the remote host used to manage the device.

### **SSH**

Telnet is commonly used to remotely manage Cisco devices. Telnet is inherently insecure and the reason for this insecurity is that all Telnet messages are sent in plain text including configuration commands and even usernames and passwords in those configuration commands. All what an attacker has to do is to be able to sniff this communication and then he owns your network. Once your network devices are compromised, they can be used as launching pad for attacking more interesting systems such as servers.

One effective alternative to this inherent lack of security in Telnet is the Secure Shell (SSH) protocol. SSH uses secure tunnels established over an insecure network to exchange information. SSH is a client server application and your Cisco device can be configured to serve both as SSH server and client. However, a Cisco device is usually configured as SSH server to accept incoming SSH connections from a remote management station. SSH has two major versions that are referred to as SSH-1 and SSH-2. The standard TCP port **22** has been assigned for SSH and SSH servers listen on this port for incoming connections.

Just like Telnet, you can use SSH to remotely connect to a Cisco device and enter IOS commands or copy files over the network. SSH uses encrypted messages so network



communications are secure. PuTTY is a popular and free Telnet/SSH client that is available for both Windows and Linux platforms.

A Cisco router has to be configured with hostname and domain name before initial SSH configuration. The configuration goes something like this:

SSH has also to be enabled on vty lines before the router starts accepting SSH connections for remote management:

### SNMP

Simple Network Management Protocol (SNMP) is used by enterprises to manage and monitor a large number of network devices. SNMP has several uses, from monitoring and generating alerts to device configuration.

There are three main versions of SNMP:

- **Version 1:** This version is defined in RFC 1157 and simple security based on SNMP communities.
- **Version 2c:** This version is defined in RFCs 1901, 1905, and 1906 and it also uses community-based security.

- **Version 3:** This version is defined in RFCs 3413 thru 3415 and introduces a new security model supporting message integrity, authentication and encryption.

The community-based security model used by SNMP versions 1 and 2c is a known security vulnerability because of its lack of encryption and authentication. It just uses a simple community name for security. Configuration of SNMP version 3 is more complex, and it should be preferred for enhanced security especially when traffic has to be moved across untrusted networks.

### Syslog

Syslog is a method that can be used to collect system messages from Cisco devices to a system running a syslog server. All system messages are sent to the central syslog server which helps in aggregation of logs and alerts. Cisco devices can send their logging messages to a Unix-like SYSLOG service. A SYSLOG service simply accepts log messages, and stores them in files or prints them according to a configuration file. Syslog uses UDP as its transport protocol and listens on port 512. This form of logging is the best available for Cisco devices because it can provide external long-term storage of logs. But this external storage of logs can be useful in incident handling when a device is compromised or undergoes a crash.

These logs are also useful in routine maintenance activities and the timestamps with each log message provide an accurate chronological record of important events happening in your Cisco device. But in order to make these timestamps meaningful, the time on your network devices must be accurate and synchronized to the same source. Network Time Protocol (NTP) is typically used to make sure timing information in Syslog messages is accurate. Network devices can use NTP to synchronize their clocks to a central accurate source of timing information.

### Network Time Protocol (NTP)

Network Time Protocol (NTP) is used to synchronize the time on the Cisco device clock. NTP usually gets its time from an accurate and trusted time source, such as a radio clock or an atomic clock attached to a time server. NTP is a client server protocol and

uses UDP port 123 as both the source and destination. NTP communications can be secured using an authentication mechanism that uses the MD5 algorithm.

NTP is absolutely essential for syslog messages as it is used to keep accurate timing information. Timestamps with syslog messages have to be accurate to make the logging information useful for troubleshooting or incident handling. The Cisco IOS **ntp** command is used in global configuration mode for all NTP related configurations.

## **8-6 Secure Communications**

Encryption techniques are commonly used at all layers of the OSI reference model to ensure security of network communications.

### **IPsec**

IPsec (Internet Protocol Security) VPN is a standard defined by the IETF (Internet Engineering Task Force). IPsec is a popular framework used to secure communications over an insecure medium like the Internet at the network layer of the OSI reference model. IPsec uses a combination of various techniques to provide the following security services:

- Peer authentication
- Data confidentiality
- Data integrity

IPsec has two methods of propagating the data across a network:

- **Tunnel Mode:** This IPsec mode is used in network-to-network or site-to-site scenarios. Tunnel mode encapsulates and protects the whole IP packet including the original IP header and payload. It then adds the IPsec header along with a new IP header as well.

- **Transport Mode:** This IPsec mode is used in host-to-host scenarios only. In transport mode, IPsec protects only the payload of the original IP packet by excluding the IP header and inserts the IPsec header between the original IP header and the payload. Transport mode is available only when the IPsec endpoints are themselves the source and destination of IP packets.

Both IPsec tunnel mode and transport mode can be deployed with Encapsulating Security Payload (ESP) or Authentication Header (AH) protocols.

## SSL

SSL (Secure Sockets Layer) is a remote access VPN technology that provides secure connectivity from any computer through a standard web browser and its native SSL encryption.

SSL is an application layer (layer 7) cryptographic protocol that provides secure communications for web browsing, email, instant messaging, and other traffic over the Internet. By default, SSL makes use of TCP port 443.

The major advantage of SSL VPN is that it does not require a special client software to be installed on the system. SSL uses the native SSL encryption of a web browser enabling a user to connect from any computer, whether it is an official desktop or a personal laptop, tablet or smartphone.

The Cisco Remote Access VPN solutions offer both IPsec VPN and SSL VPN technologies on a single platform such as Cisco Integrated Services Routers (ISRs).

## Summary

It was an introductory chapter that attempted to provide you a glimpse into the exciting world of network security. We started the chapter by talking about the CIA triad and how it can be used as a model to secure data and systems.

We also considered what information security threats are faced by enterprises today and what a typical secured enterprise looks like at a high level. We considered some layer 2 security techniques moving on to discuss a few examples of securing the management and data planes.

IPsec and SSL were briefly touched though these topics would be covered in greater detail in a later chapter.

## **Chapter 9 – Access Lists**

Cisco IOS Software has several built-in security tools that can be used as part of a good overall security strategy which are covered on the CCNA exam. Probably, the most basic of those security tools are access control lists (ACL) or access lists. Access lists enable us to identify *interesting* traffic by providing the basic capability to match packets based on a number of criteria. The *interesting* traffic can then be subjected to various special operations depending upon the specific application. This chapter reviews different types of ACLs, that are available and displays examples of how each of them would be configured in operation. We also introduce Cisco Configuration Professional and how to use it to apply ACLs toward the end of the chapter.

- 9-1 Introduction to Access Lists
- 9-2 Standard Access Lists
- 9-3 Extended Access Lists
- 9-4 Access Lists -Remote Access, Switch Port, Modifying & Helpful Hints
- 9-5 Cisco Configuration Professional Initial Setup and Access List Lab

## **9-1 Introduction to Access Lists**

The technical name for an access list is access control list (ACL) and individual entries in an access control list are called access control entries (ACEs). ACEs are also known as access list statements. The term access control lists isn't often used in practice and these lists are typically referred to simply as access lists or ACLs. An access list is simply a list of conditions or statements that categorize and match packets in a number of interesting ways.

Access lists are primarily used as simple filters to permit or deny packets through interfaces in order to exercise control on traffic flowing through the network. But this is not the only use of access lists and they can also be used in situations that don't necessarily involve filtering packets. I have listed a few of those other uses of access lists here:

**Management Access** You can use access lists to control which hosts can remotely manage your router using Telnet or SSH by applying access lists to VTY lines using the **access-class** statement in line configuration mode.

**Route Advertisement** You can use access lists to control which routes or networks will or will not be advertised by dynamic routing protocols like RIP, EIGRP, or OSPF. In such situations, access lists are defined in the same manner but the difference is where you apply those access lists. When access lists are used to control route advertisements they are called distribute lists.

**Debug Output** Cisco IOS **debug** commands are very useful for deep network troubleshooting but these commands often produce a lot of output which can be difficult to read and interpret. You can define access lists to identify interesting packets and use the access lists with debug command to display only the output that relates to interesting traffic.

**Encryption** When encrypting traffic between two routers or a router and a firewall, you must tell the router what traffic to encrypt, what traffic to send unencrypted, and what traffic to drop. Access lists are a natural choice to match or identify interesting traffic for these operations.

You should be able to appreciate the variety of ways in which access lists are used. We will not cover all of these other uses of ACLs in this chapter. However, you will see these uses of ACLs in more advanced Cisco certification exams as you move on in your career. Our coverage of access lists will focus on their use as traffic filters

When you're creating access lists (or any configuration, for that matter), it's a good idea to create them first in a text editor like Notepad, and then once you've worked out all the details, try them in a lab environment. Keep in mind that access lists are traffic filters applied to interfaces and anytime you're working on filters, you risk causing an outage to a production network.

As you have learned, access lists are the means whereby Cisco devices categorize and match packets and have several applications. The good news here is that regardless the specific application of access lists, they are defined the same way. Access lists are a very important topic for your CCNA exam so we will go into great depth while covering access lists in the next several sections.

## Access Lists Statements

An access list is basically a sequential listing of statements also known as access control entries (ACEs). Each entry in an access list defines a specific condition that packets are compared against before taking the specified action. Each access list statement specifies a **permit** or **deny** action to be taken if a packet matches the associated condition. Please see Table 9-1 for a few simple examples of access list statements:

**Table 9-1** Simple Access List Statements

Access List Statement	Description
<b>permit host 172.16.34.2</b>	Match packets with a source IP address of 172.16.34.2 only and <b>permit</b> those packets.
<b>deny host 172.16.34.24</b>	Match packets with a source IP address of 172.16.34.24 only and <b>deny</b> those packets.
<b>permit any</b>	Match and <b>permit</b> any and all packets.

There can be several **permit** and **deny** statements in an access list. A packet is compared with each statement one by one in a sequential order. That is, it'll start with the first line of the access list, then go to line 2, then line 3, and so on. If the packet matches the condition on a line of the access list, the packet is acted upon and no further comparisons take place. There is also an implicit "deny" at the end of each access list. Which means that if a packet doesn't match the condition on any of the lines in the access list, the packet will be discarded.

If you have some exposure to computer programming or scripting, you would be able to appreciate that access lists are much like a series of **if-then** statements found in many programming languages. When a given condition in the **if-then** statement is met, then a given action is taken. If the specific condition isn't met, no action is taken and the next statement is evaluated.

## Named versus Numbered



Access lists on Cisco devices can be either named or numbered. Named access lists are referenced with a name such as **UET** or **CertificationKits**. Numbered access lists are the older method, where each ACL is defined by a number such as **1** or **104**. In practice, both numbered and named access lists are widely used but I personally believe named access lists make your configuration more readable and less cryptic. Some devices such as certain Cisco Nexus switches don't support numbered access lists at all. I would advise using named access lists in the real world where possible, but for the sake of your CCNA exam you should be thoroughly familiar with both formats.

### What are Wildcard Masks?

Wildcard masks, also known as inverse masks, are used in many devices for creating access lists. Wildcards are used with IP addresses in IP access lists to specify a single host, a network, a subnet, or a supernet in order to control what should be permitted or denied. These masks are typically written in dotted decimal notation just like regular IP addresses and are quite confusing at first simply because they're the opposite, in binary, of subnet masks.

Subnet masks used while configuring IP addresses on interfaces start with 255 and have the large values on the left side, for example, IP address 192.168.2.29 with subnet mask 255.255.255.0. Wildcard masks for IP access lists are the reverse, for example, 0.0.0.255. In other words, the wildcard mask you would use to match a range that is described with a subnet mask of 255.255.255.0 would be 0.0.0.255.

When the value of a wildcard mask is broken down into binary (0s and 1s), the results determines which address bits (binary digits) are to be considered in processing the traffic, and which address bits are to be ignored. A 0 in a wildcard mask indicates that the corresponding address bit must be considered (matched); a 1 in the wildcard mask is a "don't care" meaning that the value of the corresponding address bit doesn't matter. The following table further explains the concept.

**Table 9-2** Wildcard Mask Example

Value	Explanation
-------	-------------

192.168.2.0	Network address
255.255.255.0	Subnet mask
0.0.0.0.255	Wildcard mask
11000000.11001000.00000000.00000000	Network address (binary)
11111111.11111111.11111111.00000000	Subnet mask (binary)
00000000.00000000.00000000.11111111	Wildcard mask (binary)

For class A, B, and C subnet masks, the conversion to wildcard masks is easily done by replacing all 0s with 255s, and all 255s with 0s.

**Table 9-3** Classful Wildcard Masks

Subnet Mask	Matching Wildcard Mask
255.0.0.0	0.255.255.255
255.255.0.0	0.0.255.255
255.255.255.0	0.0.0.255

While this conversion seems pretty straightforward, in the real world, networks are often not designed on classful boundaries. As an example, consider a subnet mask of **255.255.240.0** whose corresponding wildcard mask works out to be **0.0.15.255**.

There is a simple procedure to figuring out all wildcard masks, and it is easier than you might think. Here it is using the subnet mask 255.255.240.0:

1. Replace all 0 octets in the subnet mask with 255 and all 255 octets with 0 for a result of 0.0.240.255. Do not change octets that are neither 0 nor 255.
2. Calculate the number of host addresses provided by 240 in the third octet of subnet mask 255.255.240.0 yielding 16 hosts.
3. The wildcard mask will be a derivative of the number of host addresses provided by the subnet mask minus one. As  $16 - 1 = 15$ , replace this value for the third octet not changed in step 1.
4. The resulting wildcard mask is 0.15.255.255.

If it all seems overwhelming at first don't be afraid, as the more you work with inverse masks, the easier it becomes.

**Table 9-4** Sample Wildcard Masks

Wildcard Mask	Wildcard Mask in Binary	Description
0.0.0.0	00000000.00000000.00000000.00000000	The entire IP address must match.
0.0.0.255	00000000.00000000.00000000.11111111	Just the first 24 bits or 3 octets must match.
0.0.255.255	00000000.00000000.11111111.11111111	Just the first 16 bits or 2 octets must match.
0.255.255.255	00000000.11111111.11111111.11111111	Just the first 8 bits or 1 octet must match.
255.255.255.255	11111111.11111111.11111111.11111111	Matches any and all addresses automatically.
0.0.0.1	00000000.00000000.00000000.00000001	Just the first 31 bits must match.
0.0.3.255	00000000.00000000.00000011.11111111	Just the first 22 bits must match.
0.31.255.255	00000000.00011111.11111111.11111111	Just the first 11 bits must match.

### Where to Apply Access Lists?

When access lists are used as packet filters, they can be applied to either inbound or outbound traffic on any interface. Applying an access list to an interface in a certain direction causes the router to compare every packet crossing that interface in the specified direction and then either forward or discard the packet based on **permit** or **deny** action in the access control entry (ACE) that matched the packet.

If the access list is applied inbound, when the router receives a packet, the Cisco IOS Software checks the access list's permit or deny statements for a match. If the packet is permitted, the software continues to process the packet. If the packet is denied, the software simply drops the packet. If the access list is applied outbound, after receiving and routing a packet to the outbound interface, the software checks the access list's permit or deny statements for a match. If the packet is permitted, the software transmits the packet. If the packet is denied, the software just discards the packet. However, access lists that are applied to routers interfaces *do not* filter traffic that originates from that router itself.

Access lists can be applied to an interface in either inbound or outbound direction. Figure 9-1 shows a simple router with just two interfaces, Fa0/0 and Fa0/1. I have labeled the points where an access list could be applied. Keep in mind that these directions are from the device's viewpoint. Another way to understand the device's viewpoint is as if you were standing in the center of it.

**Figure 9-1** Access List Application Points

As you have seen, an access list can be applied to an interface in either an inbound or outbound direction and the direction is from the device's viewpoint. In practice however, access lists should almost always be applied *inbound* on an interface and there are good reasons for that which we will cover later.

**Exam Concept –** An access list is created using **access-list** command but it is not effective unless it is actually applied to an interface using **ip access-group** command. On the CCNA exam, you will generally be asked to apply an access-list to an interface. Make sure you select the **ip access-group** command.

When you are trying to filter traffic you usually want to prevent it from getting into the network or to a device. Applying access lists to the inbound side of an interface keeps the packets from entering the device, thus saving processing time. When a packet is allowed into a device, then switched to another interface only to be dropped by an outbound access list, the resources used to switch the packet have been wasted.

You can configure standard or extended access lists on any router in your network. But it makes more sense to place standard access lists as close to the destination as possible. It is so because a standard access list can only filter on the basis of source IP address. If it is placed near the source, you may block the source IP address for your entire network rather than for a smaller portion of your network. Extended access lists provide more granular control and you can specify exactly what you want to filter. By placing extended access lists near the source you will conserve bandwidth and router resources.



**Exam Concept** – A standard access list should be placed close to the destination while an extended access list should be placed close to the source of traffic being filtered. Where each type of access list is placed is a common CCNA question.

### Access List Logging

Access list logging is accomplished using the optional **log** keyword used with the **access-list** command when the access list is created. The **log** keyword causes an informational logging message about the packet that matches the entry to be sent to the console. The log message includes the access list number, whether the packet was permitted or denied, the source address, and the number of packets. As a large number of packets would typically match an access list, the message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets permitted or denied in the last 5-minute interval.

### Access List Remarks

You can include comments or remarks about individual entries in a named IP access list. An access list remark is simply an optional comment before or after an access list entry that describes the entry in plain language, so you don't have to interpret the purpose of the entry by its command syntax.

The remark can go before or after a **permit** or **deny** statement, but you should be consistent about where you put your remarks so that it is clear which remark describes which statement. It could be confusing to have some remarks before the associated **permit** or **deny** statements and some remarks after the associated statement.

### Types of Access Lists

Cisco IOS Software supports the following types of access lists for Internet Protocol (IP):

**Standard Access Lists** Standard access lists use source IP addresses for matching packets.

**Extended Access Lists** Extended access lists use source and destination IP addresses for matching packets and optional protocol type information for finer granularity of control.

**Reflexive Access Lists** Reflexive access lists allow IP packets to be filtered based on session information. Reflexive access lists contain *temporary* entries, something not found in standard and extended access lists, and are nested within extended named access lists.

**Time-based Access Lists** Time-based access lists, as the name indicates, are not active all the time but rather are triggered by a time function.

**Table 9-5** Packet Matching Criteria for Access Lists

Packet Matching Criteria	Standard ACL	Extended ACL
Source IP address	Yes	Yes
Destination IP address	No	Yes
Protocol (TCP, UDP, ICMP, EIGRP, OSPF etc.)	No	Yes
TCP/UDP Source/Destination Port	No	Yes
QoS parameters (ToS, IP Precedence, DSCP)	No	Yes

## **9-2 Standard Access Lists**

Standard access lists are the oldest type of access lists, dating back as early as Cisco IOS Software Release 8.3. Standard access lists control traffic by comparing the source address of packets to the addresses configured in the access list.

In all software releases, the access list number for the standard IP access lists can be anything from 1 to 99. In Cisco IOS Software Release 12.0.1, standard IP access lists began using additional numbers from 1300 to 1999. These additional numbers are sometimes referred to as the *expanded range*. In addition to using numbers to identify access lists, Cisco IOS Software Release 11.2 and later added the ability to use names to define standard IP access lists. We will learn how to configure both numbered and named access lists as we proceed in this chapter.

**Table 9-6** Access List Types and Corresponding Numbers

Access List Type	Number Range
IP Standard Access Lists	1-99
IP Standard Access Lists (expanded range)	1300-1999
IP Extended Access Lists	100-199
IP Extended Access Lists (expanded range)	2000-2699

You can differentiate between standard and extended access lists in the numbered format simply by looking at the access list number. Based on the number used when access list is created, the router also knows which type of syntax to expect as the list is entered. By using numbers 1 – 99 or 1300 – 1999, you are essentially telling the router that you want to create a standard IP access list. Thus the router will expect the standard IP access list syntax specifying only the source IP address in access list entries.

### Creating a Numbered Standard Access List

If you want to filter traffic using source IP address only, a standard access list is a simple and sufficient option. Probably the best way to show you how to configure a numbered standard access list is by showing you how to do it one step at a time:

The command to create an access list, not surprisingly, is **access-list** entered in configuration mode. As we just discussed the number we use to identify an access list cannot be any arbitrary number. This number rather must belong to the range of numbers available for the type of access list you want to create. At the moment, we are interested in creating a standard numbered access list. So we can choose a number from the ranges 1-99 or 1300-1999.

We chose to use 1 as our standard access list number and there are three keywords available now as you can see from above output. Let's first add a user-friendly remark in order to make our access list more readable as we return to it at a later point in time. A remark of upto 100 characters can precede or follow an access control entry. We will add a remark *before* the entry though you can choose to add remarks following access list entries. You should be consistent throughout your configuration whether you choose to add remarks before or after access control entries.

As you may have guessed from the remark, we are going to create an access control entry that would allow all traffic sourced from Bob's IP address 172.16.23.3. We use the wildcard mask 0.0.0.0 in order to match all bits in the source IP address. If the source wildcard mask is omitted, a wildcard mask of 0.0.0.0 is assumed. So, in this case we may even have omitted the wildcard mask of 0.0.0.0 but it is good practice to always explicitly mention the wildcard mask to make configuration readily understandable. Keep in mind that good configuration is one that is easy to read and understand, not one that is cryptic and unnecessarily complex.



Exactly the same result can be achieved using the **host** keyword.

We can use either of the two formats just described to have exactly the same effect. Let's move on to create our next access control entry denying access to Max.

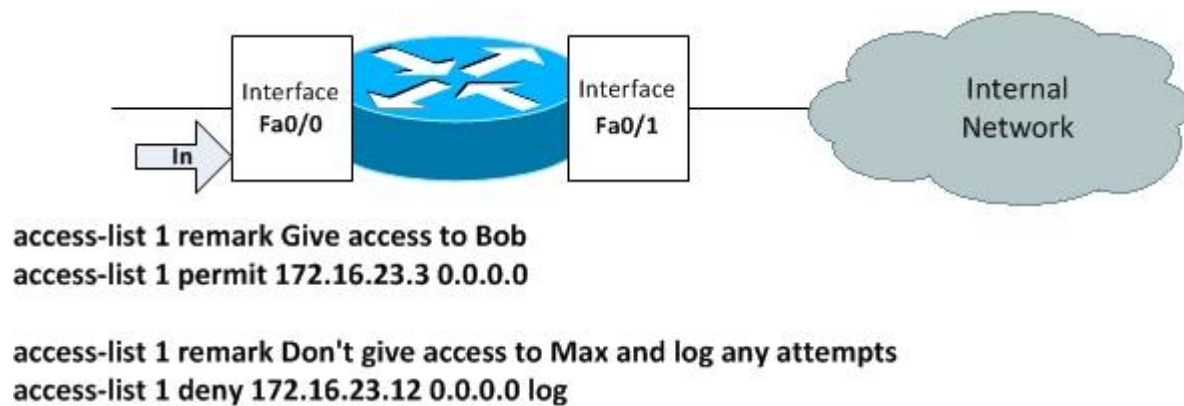
Now that's interesting as we intend to not only deny access to Max but also to log any access attempts made by him using the **log** keyword at the end of the statement.

At this point, the access list has actually been created and the same can be verified. This access list would be visible in the running configuration of the router when you use the command **show running-config** using the **include** keyword to filter output showing only those lines in the running configuration that contain the *access-list* keyword.

Another useful command to verify access lists is **show access-lists**:

Though the access list has been created now but it is sitting idle, doing nothing as it has actually not been *applied* to any interface. Let's go ahead and apply it to interface Fa0/0 in the *inbound* direction as depicted in Figure 9-2.

**Figure 9-2** Standard Numbered Access List Example



The command to apply an access list to an interface is **ip access-group** entered in interface configuration mode:

As you can see, there are two options available while applying an access list to an interface, those are, **in** and **out**. We have applied the access list in the inbound direction filtering packets coming into the interface from outside. The access list is now applied comparing all packets received on interface Fa0/0 against entries in access list 1 and taking appropriate action.

Let's run a final check verifying if the access list has been successfully applied to the router interface:

### Creating a Named Standard Access List

A standard named access list can be used if you need to filter on source address only. There is no difference between numbered and named access lists in terms of functionality, however each has its own syntax. We will define a standard named access list including one **permit** statement and one **deny** statement. The actual statements you use and their order would depend on your filtering requirements. You should define your **permit** and **deny** statements depending on what you want to allow or block.

Enter privileged exec mode using **enable** command, and move to the global configuration mode using **configure terminal** command. You can configure access lists from the global configuration mode just like any other configuration on Cisco devices.

The command used to define a named access list is **ip access-list** which has several options:

We are specifically interested here in two of these options: **standard** and **extended**, used respectively to defined standard and extended access lists. We will proceed to define and standard access list named **Corp**.

Let's now define access list statements preceded by remarks to make their meaning clear. The first access list entry denies the whole class B network 172.18.0.0 belonging to the Operations department and also logs any unsuccessful attempts using the log keyword.

But the CEO has an IP address 172.16.3.24 and he still needs to have access, so we create a **permit** statement to do just the same.

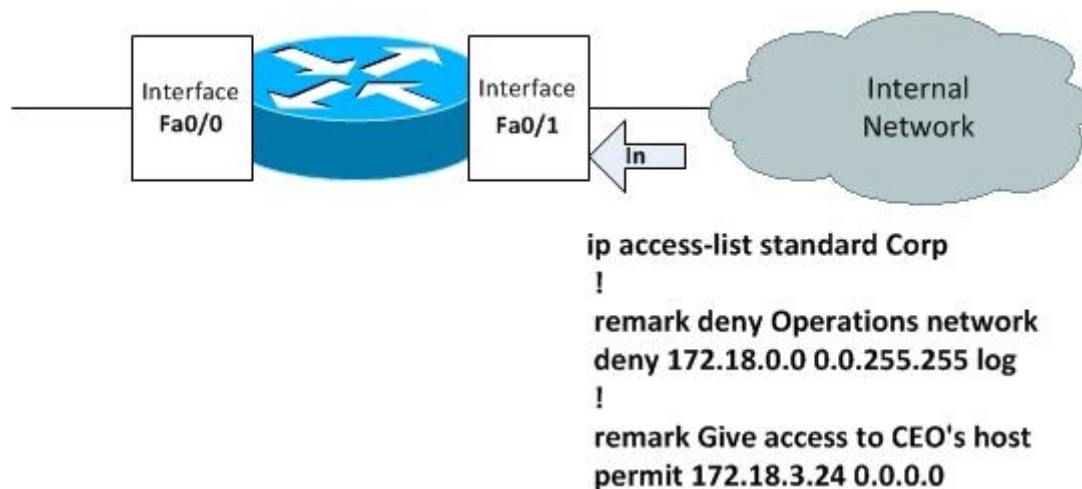
And that's all! The access list has been created which can be verified using command **show ip access-list**.



**Exam Concept** – Use the **show ip access-list** command to verify an access list has been created, while use the **show ip interface** command to verify that the access list is applied to an interface. Cisco now wants you to thoroughly understand how to use the show commands for troubleshooting on the CCNA

And let's have a look at the figure before applying the access list to an interface.

**Figure 9-3** Standard, Named Access List Example




Let's now finalize our configuration by applying the access list to router interface Fa0/1 in the inbound direction.

You can now verify if access list has actually been applied to the interface Fa0/1 using command **show ip interface**.

## **9-3 Extended Access Lists**

Standard access lists are sufficient if you want to filter on source IP address only. But if you want to filter on anything other than source address, you would have to configure an extended access list: numbered or named. Extended access lists can filter on source and destination IP addresses, or a combination of addresses and several other fields. If you prefer not to use a name, extended IP access lists can be numbered 100 – 199 or 2000 – 2699.

### **TCP/UDP Packet Matching**

Standard access lists are *protocol aware* which means they can be used to match packets on THE BASIS OF  layer 4 protocol. As you can see in the output below an extended access list can match packets on the basis of TCP, UDP, ICMP, EIGRP, and OSPF. In fact, you can even specify the protocol number found in the IP header to identify the higher layer protocol. It is useful if a keyword for the protocol you are trying to match is not available on your specific Cisco IOS Software release. You can find a

complete list of assigned IP protocol numbers at  
<http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml>.

**Table 9-7** Some Protocols in the Protocol Field of IP Header

Protocol	Protocol Number
ICMP	1
TCP	6
UDP	17
IPv6	41
GRE	47
EIGRP	88



OSPF	89
PIM	103

You should especially be careful which layer 4 protocol to use when trying to match packets in extended access lists. For example, if you are trying to match telnet traffic, you would use **tcp** rather than **udp** keyword as shown below. Other examples of protocols running on TCP include Hyper-text Transfer Protocol (HTTP), File Transfer Protocol (FTP), and many more.

You may need to use **udp** keyword if the protocol you are trying to match runs on UDP. Examples of such protocols are Routing Information Protocol (RIP), Trivial File Transfer Protocol (TFTP), and Network Time Protocol (ntp) to name a few.

Standard access lists for TCP have another interesting matching criterion available in the form of **established** keyword.

You may recall that TCP uses a three-way handshake as part of TCP connection establishment. It is called three-way handshake because three TCP messages (SYN,

SYN-ACK, and ACK) are exchanged between two hosts to establish a TCP connection. SYN and ACK each refers to a specific bit in the TCP header. The ACK bit *is not* set in the SYN message, the very first TCP message exchanged. However, the ACK bit *is* set in the other two messages of the three-way handshake as well all subsequent TCP messages. The **established** keyword simply tracks the ACK bit in the TCP header. An access list entry with **established** keyword matches all packets in a TCP session (provided other criteria are also met) other than the very first packet in which ACK bit is not set. But what's the use of tracking this ACK bit? Let's assume we apply an access list *inbound* to the Internet facing interface of our router, making use of the **established** keyword. It would prevent any inbound TCP connections from the Internet to your internal systems because the very first TCP message would be blocked. TCP connections from inside to the Internet would still be allowed. It acts as a basic security mechanism to protect your network.

### ICMP Packet Matching

Internet Control Message Protocol (ICMP) provides messaging services to IP and works at the network layer. ICMP messages carried inside IP packets provide information about network problems.

**Table 9-8** ICMP Message Types

Keyword	Description
echo	Echo request (used to ping)
echo-reply	Echo reply (used to ping)
host-unreachable	The packet was delivered to the destination network but could not be sent to the specific host with destination IP address.
net-unreachable	The packet could not be delivered to the destination network. It usually indicates a routing issue.
port-unreachable	The destination port specified in the TCP or UDP header was invalid for the host to which the packet was sent.
protocol-unreachable	The protocol specified in the IP header was invalid for the host to which

	the packet was sent.
ttl-exceeded	Time-to-live (TTL) expired while packet was in transit.

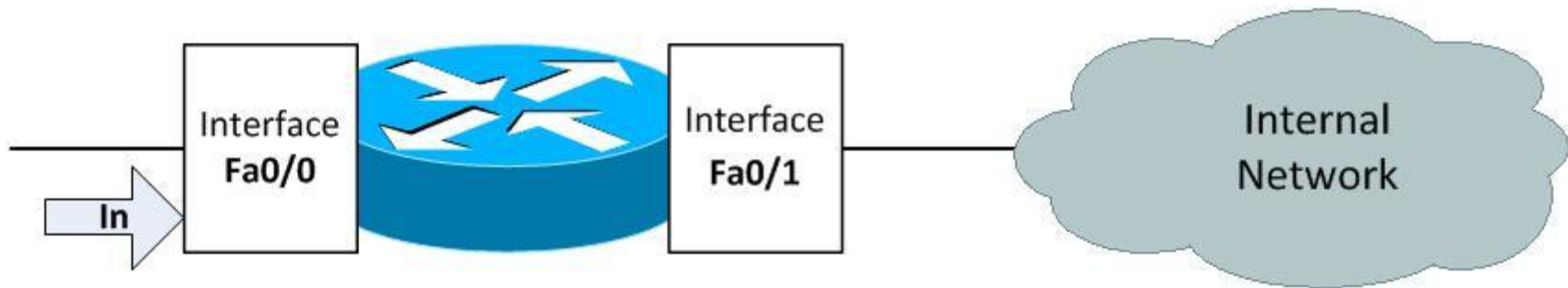
Extended access lists can be used to match specific ICMP message types based on protocol number of several keywords available on the Cisco IOS. You can find out which keywords are available as shown below in abridged output from Cisco CLI:

Access list statements can use either one of available keywords or numeric ICMP message type to match specific types of ICMP packets. A complete list of ICMP message types can be found at <http://www.iana.org/assignments/icmp-parameters/icmp-parameters.xml>.

### Creating a Numbered Extended Access List

Extended access lists can be created using a number in the 100 – 199 or 2000 – 2699 range. In terms of functionality, numbered and named extended access lists can be used to achieve the same results; however they have differences in syntax.

**Figure 9-4** Extended, Numbered Access List Example



**access-list 101 remark allow Telnet packets from any source to network 172.16.0.0**  
**access-list 101 permit tcp any 172.16.0.0 0.0.255.255 eq telnet**

**access-list 101 remark deny all other TCP packets**  
**access-list 101 deny tcp any any log**

If you want to filter traffic using any criteria other than source IP address, an extended access list is needed. We will now show you how to configure a numbered extended access list one step at a time:

At the moment, we are interested in creating an extended numbered access list. So we

can choose a number from the ranges 100-199 or 2000-2699.

We have chosen 101 as our extended access list number and there are three keywords available now, as you can see from above output. Let's first add a user-friendly remark as usual in order to make our access list more readable as we return to it at a later point in time.

As you may have guessed from the remark, we are going to create an access control entry that would allow Telnet traffic sourced from anywhere destined for the network 172.16.0.0. We use the wildcard mask 0.0.255.255 for our class B network 172.16.0.0.

We used the telnet keyword to specify the application. However we could also use the port number 23 reserved for Telnet to achieve the same result. The alternate configuration would look something like the output below:

Let's move on to create our next access control entry denying all other TCP packets creating a remark first as usual.

We intend to not only deny access all other TCP packets, but also to log any such packets using the **log** keyword at the end of the statement.

At this point, the access list has actually been created and the same can be verified. This access list would be visible in the running configuration of the router when you use the command **show running-config** using the **include** keyword to filter output showing only those lines in the running configuration that contain the *access-list 101* arguments.

Another useful command to verify access lists is **show access-lists**:

Though the access list has been created now, it is sitting idle doing nothing. As it has actually not been *applied* to any interface to have functionality. Let's go ahead and apply it to interface Fa0/0 in the *inbound* direction as depicted in Figure 9-4.

The command to apply an access list to an interface is **ip access-group** entered in interface configuration mode:

We have applied the access list in the inbound direction filtering packets coming into the interface from outside. The access list is now applied comparing all packets received on interface Fa0/0 against entries in access list 101 and taking a permit or deny action as appropriate.

### Creating a Named Extended Access List

A standard, named access lists can be used if you need to filter on source and destination IP address or a combination of addresses and other fields. There is no difference between numbered and named access lists in terms of functionality; however each has its own syntax.

We will define an extended named access list including one **permit** statement and one **deny** statement. The actual statements you use and their order would depend on your filtering requirements. You should define your **permit** and **deny** statements depending on what you want to allow or block.

Let's first enter the privileged exec mode using **enable** command, and move to the global configuration mode using **configure terminal** command.



The command used to define a named access list is **ip access-list** for both standard and extended access lists:

We are interested here in the option **extended** used to defined extended access lists. We will proceed to define an extended access list named **NoSales**.

Let's now define access list statements preceded by remarks to make their meaning clear. The first remark reminds the network administrator that the following entry denies access to the Sales network.

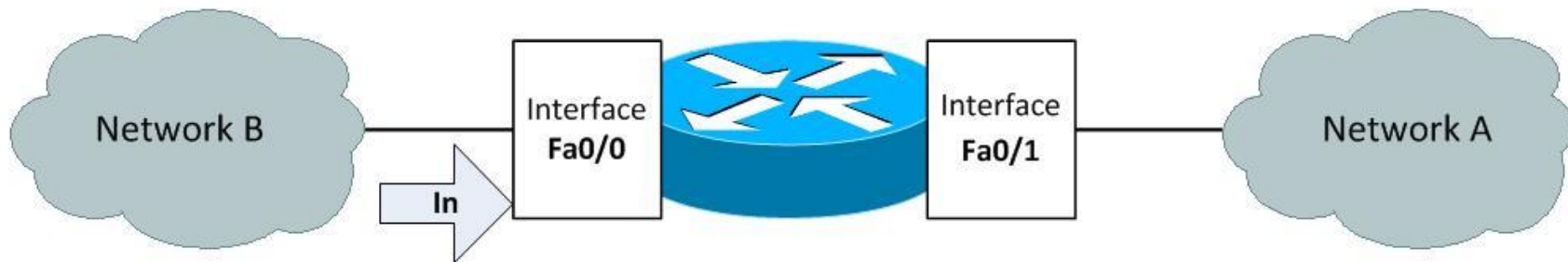
In the first access list statement, packets from the Sales network 172.18.0.0 are denied access to host 172.16.40.10 also logging all such packets.

We have to add another access list statement that allows all TCP packets from any source to any destination. We would also add a remark preceding this access lists statement as usual.

Remember that all sources not specifically permitted are denied by an implicit **deny** statement at the end of the access list.

And let's have a look at the figure before applying the access list to an interface.

**Figure 9-5** Extended, Named Access List Example



**ip access-list extended NoSales**

**!**

**remark deny access from the Sales network to the server**

**deny ip 172.18.0.0 0.0.255.255 host 172.16.40.10 log**

**!**

**remark allow TCP from any source to any destination**

**permit tcp any any**

Finally, let's apply the access list to interface Fa0/0 in the inbound direction as shown in the figure.

Finally, we can display the access list using the good old **show access-list** command.

This completes our coverage of access lists full with examples of standard and extended access lists in the numbered and named format. We also provided a bunch of hints that will help you create access lists in practice.

## **9-4 Access Lists -Remote Access, Switch Port, Modifying & Helpful Hints**

### **Restricting Remote Access with Access Lists**

Vty or virtual terminal lines are used to allow remote access to the router. A virtual terminal line is *not* associated with the auxiliary or console port. The router has five virtual terminal lines by default numbered 0 through 4. You can create additional virtual terminal lines if more than five concurrent remote console connections are desired. In most situations five default virtual terminal lines are enough.

Securing remote console access is critical because if it is compromised, an intruder can gain access to router configuration and can even modify it which compromises all other security features configured on the router. Vty lines are also protected using usernames and passwords but access lists can also be used as an additional security measure

ensuring Vty lines can be accessed only from trusted hosts having specific IP addresses. Also remember to set identical access lists on all active virtual terminal lines because a user can connect to any of them.

The **access-class** command is used in line configuration mode to restrict incoming or outgoing connections between a virtual terminal line and the addresses in the access list. The following example defines an access list that denies incoming connections from all networks other than 172.16.0.0 on terminal lines 0 through 4.

The **show line** command can be used to view at a glance all active virtual terminal lines and access lists applied to them.

## Modifying Access Lists

While you are creating an access list or after it is created, you might want to delete an entry. You cannot delete an individual entry from a numbered access list. If you need to delete even a single entry from a numbered access list, you have to delete the whole access list using **no access-list** command and start over.

It is a good strategy to copy the access list to Notepad before deleting it from router configuration. You can then modify the access list in Notepad before applying it again to router configuration.

However, you sure can delete an individual entry from a named access list using the **no permit** or **no deny** command. Let's demonstrate this using the **NoSales** extended access list we created earlier, by deleting the second access list statement.

This is one good reason to prefer named access lists over numbered access lists from a practical standpoint.

### Hints for Creating IP Access Lists

- Create an access list before applying it to an interface, because if you apply a yet non-existent access list to an interface and then proceed to configure the access list, the first access control entry (ACE) is put into effect as soon as you enter it, and the implicit **deny** statement that follows could immediately block traffic causing immediate access problems.
- An interface with an empty access list applied to it permits all traffic, so that's another reason to configure an access list before applying it.
- Only one access list can be applied to an interface in each direction for any given protocol.
- All access lists need at least one **permit** statement; otherwise all packets are denied due to the implicit **deny** statement and no traffic passes at all.
- Because the software stops testing access control entries (ACEs) after it encounters the first match (to either a **permit** or **deny** statement), you will reduce the processing time and resource usage if you put statements that packets are most likely to match at the beginning of the access list. Place more frequently occurring conditions before less frequent conditions.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- Use the statement **permit any any** if you want to allow all other packets not already denied by an earlier statement in the access list. Using the statement **permit any any** at the end of an access list, in effect, avoids denying all other packets with the implicit deny statement at the end of an access list. Do not make your first access list entry **permit any any** because all traffic will get



through; no packets will reach subsequent testing and so even if there are **deny** statements after **permit any any** they will have no effect.

- While you are creating an access list or after it is created, you might want to delete an entry. You cannot delete an individual entry from a numbered access list; trying to do so will delete the entire access list. If you need to delete an entry, you have to delete the whole access list using **no access-list** command and start over. However, you sure can delete an individual entry from a named access list using the **no permit** or **no deny** command. This is one good reason to prefer named access lists over numbered access lists from a practical standpoint.
- In order to make individual entries or statements in an access list more scanable and easy to understand at a glance, you can write a helpful remark before or after any statement using the **remark** command.
- When trying to save router resources, remember that an inbound access list filters traffic before the routing table lookup. An outbound access list applies the filter conditions after the routing table lookup.

## Switch Port Access Lists

Switch port access lists are ACLs configured on physical interfaces on a switch. Here are some facts you should not forget about port ACLs:

- Port ACLs support only inbound traffic filtering.
- Port ACLs can be configured as three types: standard, extended, and extended MAC.
- Port ACLs control IP traffic through standard or extended access lists while non-IP traffic is filtered through the use of extended MAC access lists.

You are already familiar with how standard and extended access lists are created. We will learn how to create extended MAC access lists in this section. You can apply both types of filters, IP and non-IP, to a single interface, but you only get to apply one of each. If you try to apply an additional ACL of either type on an interface, the new one will override the one you had there before.

Let's define and verify an extended MAC access list:

It's now time to apply the MAC ACL to a switch interface using **mac access-group** command:

Let's try to understand what we just did. We created an extended MAC access list that we called MY\_MAC\_LIST, allowing incoming frames sourced only from a specific MAC address 00cd.38ab.4d35. This scenario makes sense if you have a desktop cabled to your switch port and you don't want any other device connected to the same port by user.

In the last example, we defined an access list that made its filtering decision based on MAC addresses. Sometimes it is desirable to make permit or deny decisions based on the protocol carried inside Ethernet frames rather than source and/or destination MAC addresses.

You can specify either an EtherType code or protocol name if a corresponding keyword for your desired protocol is available.

## **9-5 Cisco Configuration Professional Initial Setup and Access List Lab**

Cisco Configuration Professional (Cisco CP) is a software tool that can be used to configure and manage standalone network devices or groups of devices.

The advantage of Cisco CP is that it simplifies configuration of Cisco access routers through graphical user interface (GUI) based easy-to-use wizards. Among the features you can configure using Cisco CP are routing, firewall, intrusion prevention system (IPS), VPN, unified communications, WAN, and LAN. Cisco CP can also be used to monitor router status and troubleshoot WAN and VPN connectivity issues. We are going to experience firsthand in a moment all this and more that can be done using Cisco CP.

In addition to replacing command line interface (CLI) with graphical user interface (GUI) for ease of configuration, Cisco CP also provides additional tools to make router deployments more efficient. These additional tools offer a one-click router lockdown and voice and security auditing capability to check and recommend changes to router configuration. Cisco CP can also monitor status of a router and troubleshoot WAN and VPN connectivity issues.

Cisco Configuration Professional Express (Cisco CP Express) is a light weight version of Cisco CP. Cisco CP Express along with a factory default configuration file are already installed in flash memory of routers that are shipped with Cisco CP. It means you can simply unpack the device and connect a PC directly to it, and then use the pre-installed Cisco CP Express to configure the device. We will cover the full version of Cisco CP here that you can install as an application on a Windows based computer.



**Exam Concept –** Cisco Configuration Professional (Cisco CP) has replaced Cisco's Security Device Manager (SDM) as the GUI configuration solution. Cisco CP is not on the CCNA at this time. However it is on the CCNA Security exam.

Cisco Configuration Professional is due to replace Cisco Security Device Manager (SDM) over time. CCP communications are pretty secure as it uses secure protocols such as Secure Shell Protocol (SSH) and HTTPS to communicate with the devices.

Newly shipped Cisco routers do not have any configuration pre-loaded which means you have to connect a console cable to the console port and use terminal emulation software like Hyper Terminal to do initial configuration of the router. But, devices shipped with Cisco CP do have a default configuration that allows you to connect a PC to an Ethernet port on the device and start configuring it right away.

Let's start by installing Cisco CP 2.5 on a Windows based computer. You should have the installation package in the form of a file such as cisco-config-pro-k9-pkg-2\_5-en.exe which you launch to start the installation process.

**Figure 9-6** Cisco Configuration Professional Installation



The installation is pretty straightforward and takes less than a minute to complete.


You can launch the application after finalizing the installation and you may be prompted to select / manage a community of devices as the application loads. You can safely cancel this dialogue box initially and reach the main application window which for version 2.5 looks like the figure below.

**Figure 9-7** Cisco Configuration Professional Main Window

ApplicationHelp

HomeConfigureMonitor

Cisco Configuration Professional



Select Community Member:  
(No devices discovered)

Router

Switching

Switching Module

Security

Traffic Monitoring

Utilities

Flash File Management

Software Upgrade

Configuration Editor

Save Configuration to PC

Write to Startup Configuration

Home > Community View

Cisco Configuration Professional News

Date	Title
07-Dec-2011	<a href="#">Cisco Configuration Professional v2.6 is now available</a>
07-Dec-2011	<a href="#">Provide CCP Feedback</a>
07-Dec-2011	<a href="#">Cisco Configuration Professional v2.6 Release Notes</a>

Community Information

Selected community: **Community Not Selected**. Select a device from the table below. Use the buttons at the bottom to cont

Filter

0 rows retriev

IP address / Hostname	Router Hostname	Connection Type	Discovery Status

Manage DevicesDeleteDiscoverDiscovery DetailsCancel DiscoveryRouter Stat

We will set up a single device, a newly shipped Cisco 881 router, to be managed using Cisco Configuration Professional. The computer on which you just installed Cisco CP should be connected to the console port of the router through its serial port. If your computer does not have a serial port, you can use an USB/RS-232 adapter to connect to the router console. After ensuring physical connectivity, go to the *Application* menu and click on *Setup New Device....* You see a screen similar to below figure.

**Figure 9-8** New Device Setup Wizard – Step 1



http://127.0.0.1:8600/Counterpoint/CPMain.html?rand=28396 - Windows Internet Explorer

Application Help

Home Configure Monitor

Cisco Configuration Professional

### New Device Setup Wizard

- 1 - Introduction
- 2 - Configuring Device
- 3 - Configuration Summary

#### Step 1 - Introduction

##### Setting up a new device

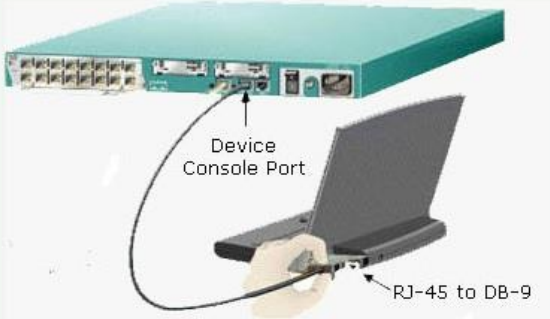
Use this wizard to setup a device to enable Cisco Configuration Professional to manage it. This wizard can be used on a device if its new, has been reset to factory default or has existing configuration. The device will be configured with authentication credentials, vty lines and an https server.

These steps are described in the [CCP Quick Start Guide](#).

Prerequisites :

1. Ensure the computer running CCP is connected to the powered up device over the console port.
2. Ensure the device baud is set to its default value
3. Ensure a community is selected

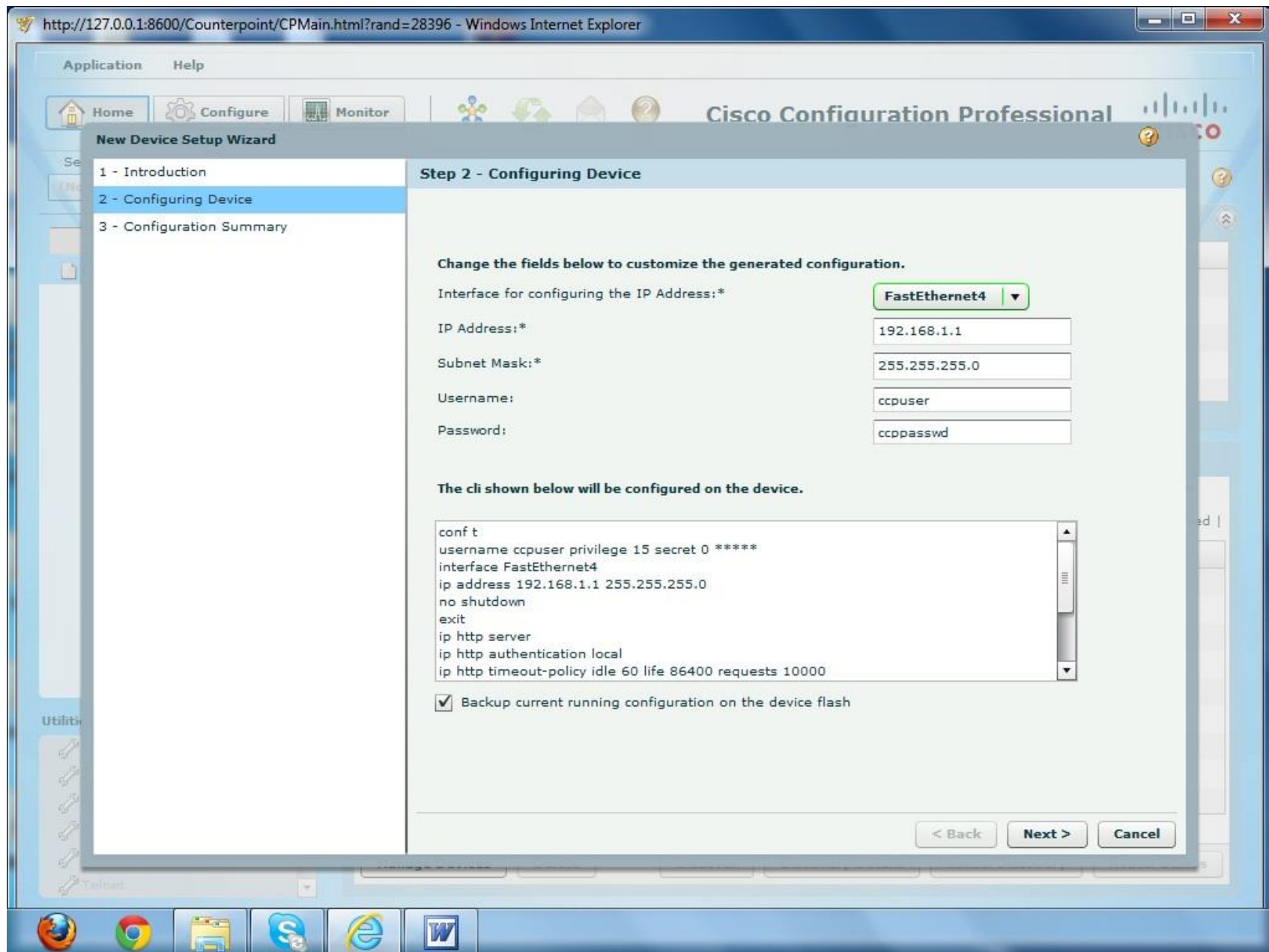
See below for a typical connection between a PC/laptop and a router. If a serial port is not available a USB to RS-232 converter may have to be used.



< Back Next > Cancel

Simply press *Next* to move to *Step 2 – Configuring Device* where you can enter IP addresses for available interfaces. In our case, we configure IP address 192.168.1.1 on interface FastEthernet4 of our Cisco 881 and press *Next*.

**Figure 9-9** New Device Setup Wizard – Step 2



If everything goes well, you reach Step 3 – Configuration Summary as shown in the figure below.

**Figure 9-10** New Device Setup Wizard – Step 3

http://127.0.0.1:8600/Counterpoint/CPMain.html?rand=28396 - Windows Internet Explorer

Application Help

Home Configure Monitor

Cisco Configuration Professional

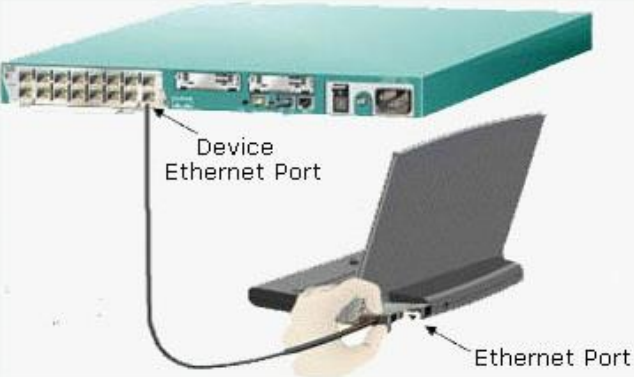
### New Device Setup Wizard

- 1 - Introduction
- 2 - Configuring Device
- 3 - Configuration Summary

#### Step 3 - Configuration Summary

**CCP has successfully configured the device. It can now be managed by CCP over an ethernet connection.**

You could now connect the PC/laptop to the router over the ethernet cable as shown below OR connect to the device over the network. If the device is connected directly as shown below, ensure that the PC/laptop is assigned an ip address in the same subnet as the device ip address



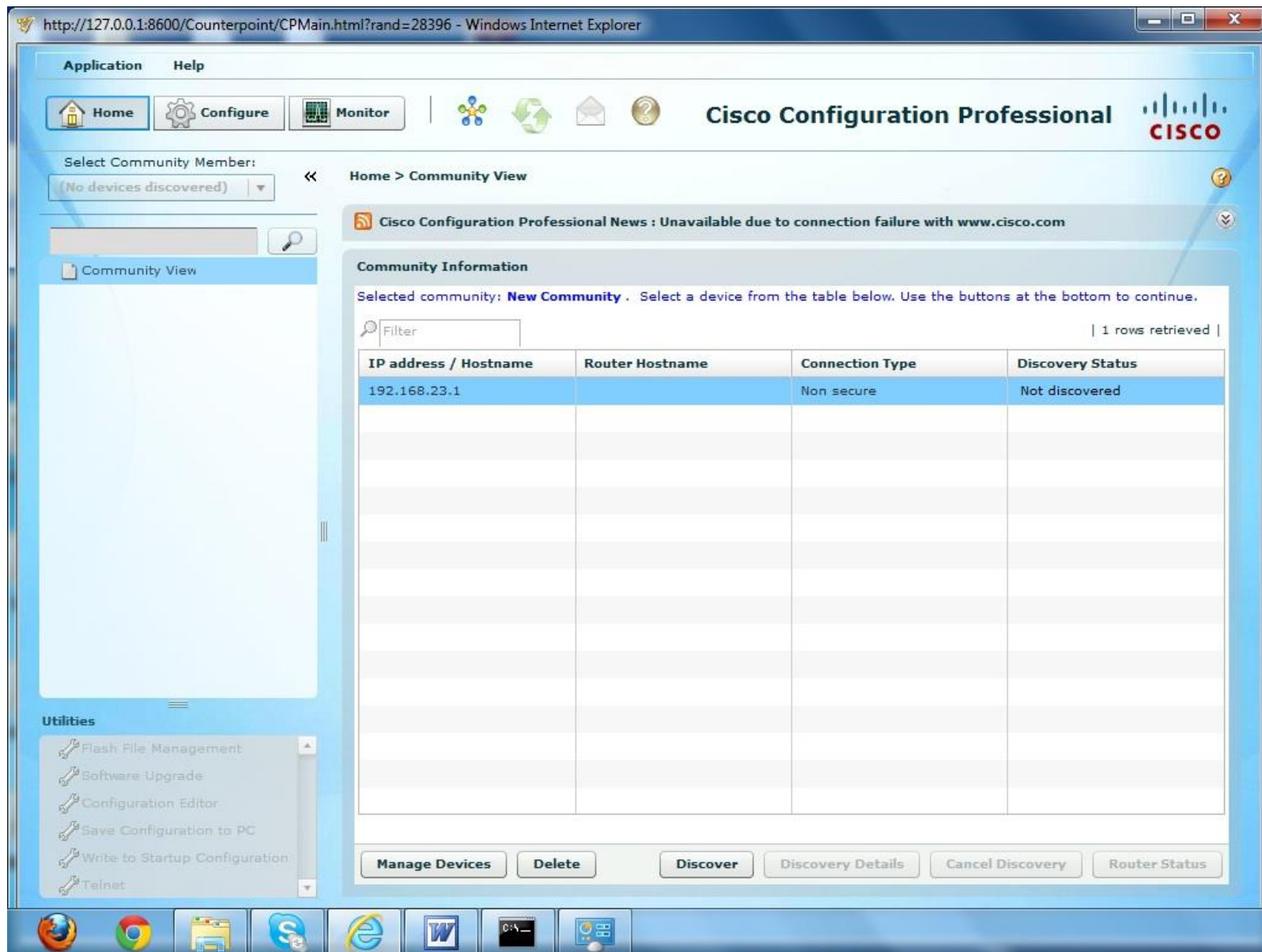
☒ Add this device to CCP's currently selected community.

< Back Finish Cancel

What we have done so far is to configure IP address 192.168.23.1 on interface FastEthernet4 of the router. Now you should connect the Ethernet port of your computer to interface FastEthernet4 of the router using a *crossover* Ethernet cable.

At this stage the main application window would look something like this:

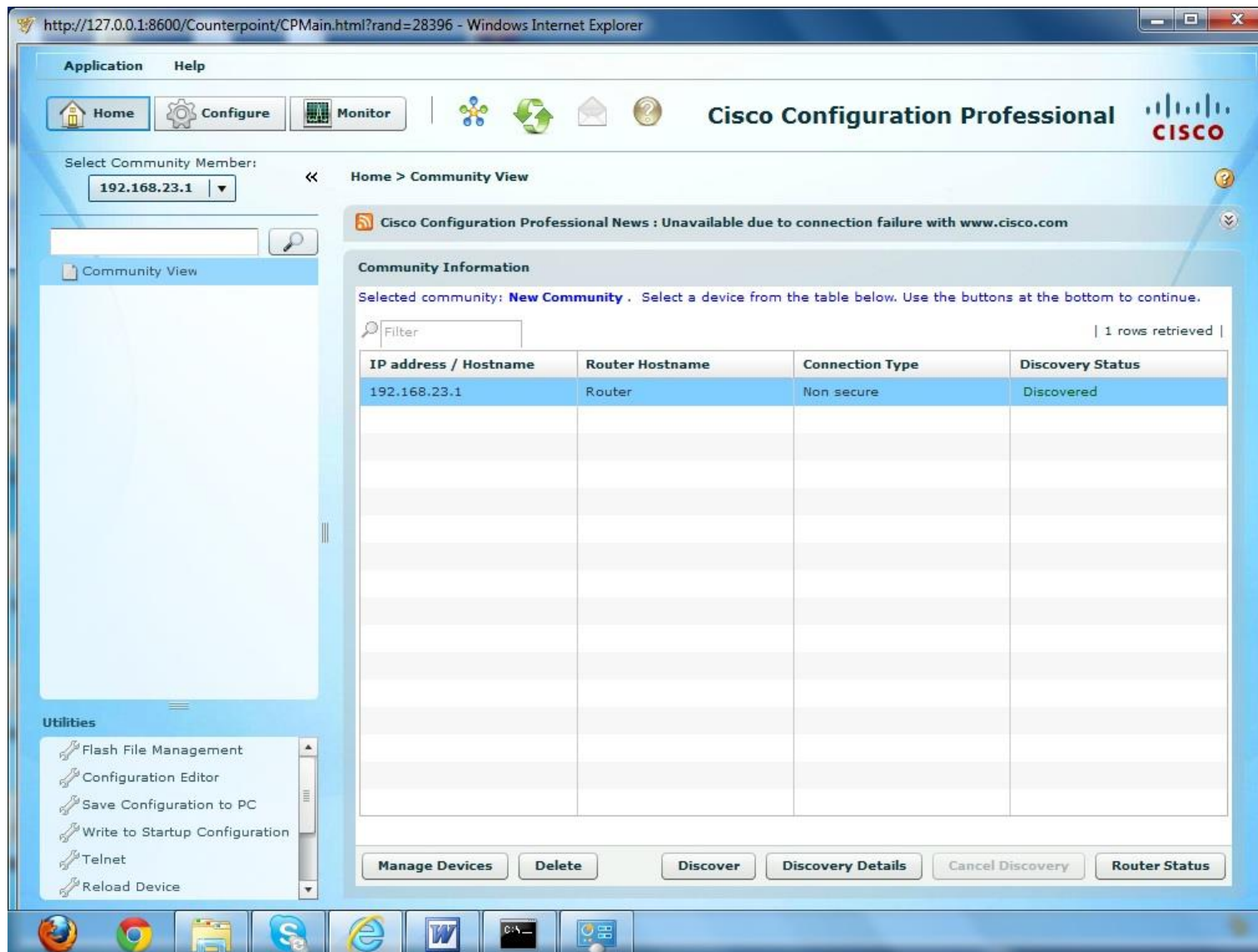
**Figure 9-11** New Device – Not Discovered



We highlight the IP address 192.168.23.1 we configured and press *Discover*. Cisco CP will not try to connect to the router over the Ethernet interface and if all goes well the *Discovery Status* should change to *Discovered* as shown in figure below.

**Figure 9-12** New Device – Discovered





At this stage the router is fully set up with Cisco Configuration Professional and we can configure it using easy-to-use wizards by pressing the *Configure* in the top left area of the display. Some new entries appear in the left pane of the display as shown in Figure 9-13.

**Figure 9-13** New Device – Configuration

http://127.0.0.1:8600/Counterpoint/CPMain.html?rand=28396 - Windows Internet Explorer

Application Help

Home Configure Monitor

Select Community Member: 192.168.23.1

Configure > Router > ACL > ACL Editor

Additional Tasks

Access Rules Add... Edit... Delete

Name/Number	Used by	Type	Description
-------------	---------	------	-------------

Action	Source	Destination	Service	Log	Attributes	Des
--------	--------	-------------	---------	-----	------------	-----

Interface Management

Router

- Router Options
- Time
- Router Access
- DHCP
- DNS
- Static and Dynamic Routing
- AAA
- ACL
  - Object Groups
  - ACL Summary
  - ACL Editor
  - NAT Rules
  - IPSec Rules
  - NAC Rules

Utilities

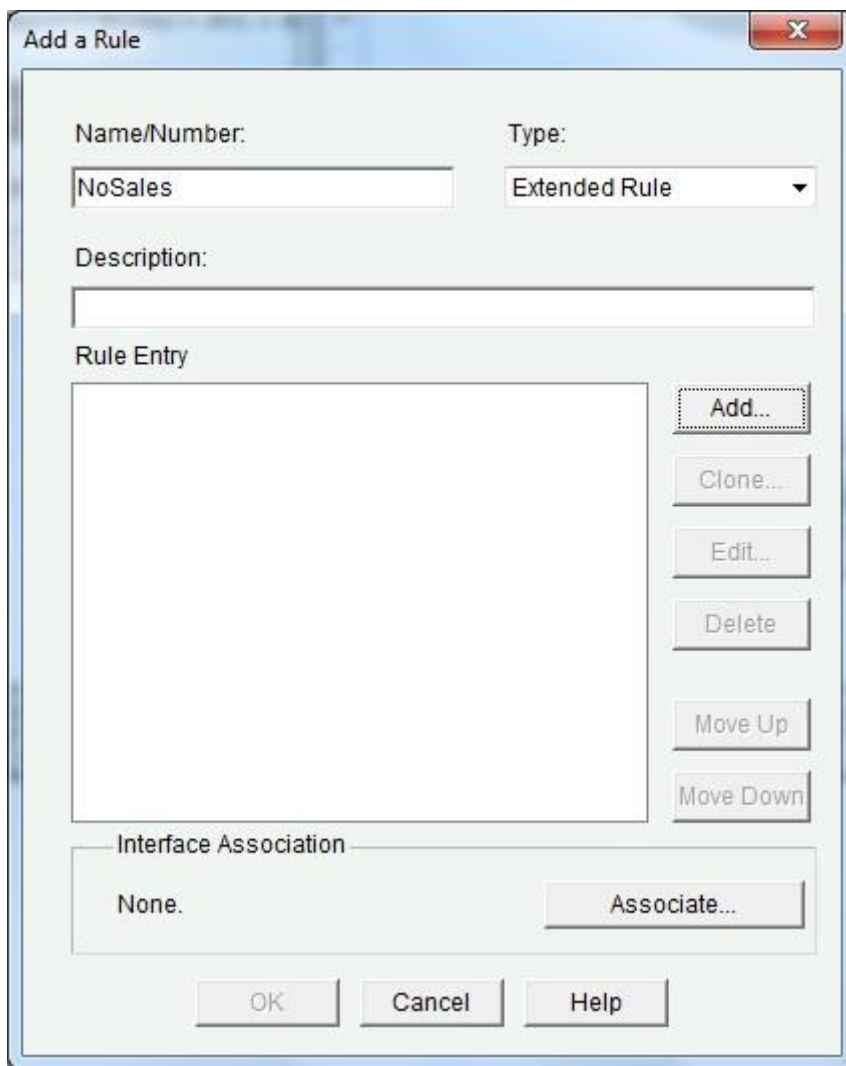
- Flash File Management
- Configuration Editor
- Save Configuration to PC
- Write to Startup Configuration
- Telnet
- Reload Device

Cisco Configuration Professional

CISCO

We will re-create the named extended access list **NoSales** this time using CCP GUI wizard. We created the same access list earlier in the chapter using command-line interface (CLI). Go to *Router > ACL > ACL Editor* in the left pane and press *Add...* to get the dialogue box shown in Figure 9-14, which can be used to enter and apply access lists as required. In this dialogue box you supply a name and specify that it is an extended ACL, and then press *Add* to create the first access list statement.

**Figure 9-14** Dialogue – Add a Rule



The "Add a Rule" dialog box is shown with the following fields and controls:

- Name/Number:** A text box containing "NoSales".
- Type:** A dropdown menu set to "Extended Rule".
- Description:** An empty text box.
- Rule Entry:** A large empty text area for entering the rule statement.
- Buttons:** "Add...", "Clone...", "Edit...", "Delete", "Move Up", and "Move Down" are located to the right of the Rule Entry area.
- Interface Association:** A section with a text box containing "None." and an "Associate..." button.
- Footer:** "OK", "Cancel", and "Help" buttons.

We now create the first access list statement as shown in Figure 9-15 and press *OK* to proceed.

**Figure 9-15** Dialogue – Add an Extended Rule Entry 1

**Add an Extended Rule Entry**

**Action**  
Select an action: **Deny**

**Description**  
deny access from the Sales network to the serv

**Source Host/Network**  
Type: **A Network**  
IP Address: 172.18.0.0  
Wildcard Mask: 0.0.255.255  
( Mask bit 0 - Must match )  
( Mask bit 1 - Don't care )

**Destination Host/Network**  
Type: **A Host Name or IP Address**  
Host Name/IP: 172.16.40.10

**Protocol and Service**  
☐ TCP ☐ UDP ☐ ICMP ☒ IP ☐ Service Object Group  
IP Protocol: ip

☒ Log matches against this entry

**OK** **Cancel** **Help**

In the same fashion we create the second access list statement as shown in Figure 9-16.

**Figure 9-16** Dialogue – Add an Extended Rule Entry 2

**Add an Extended Rule Entry**

**Action**  
Select an action: Permit

**Description**  
allow TCP from any source to any destination

**Source Host/Network**  
Type: Any IP Address

**Destination Host/Network**  
Type: Any IP Address

**Protocol and Service**  
☒ TCP ☐ UDP ☐ ICMP ☐ IP ☐ Service Object Group

**Source Port (Rarely changed. See help)**  
Service = any

**Destination Port**  
Service = any

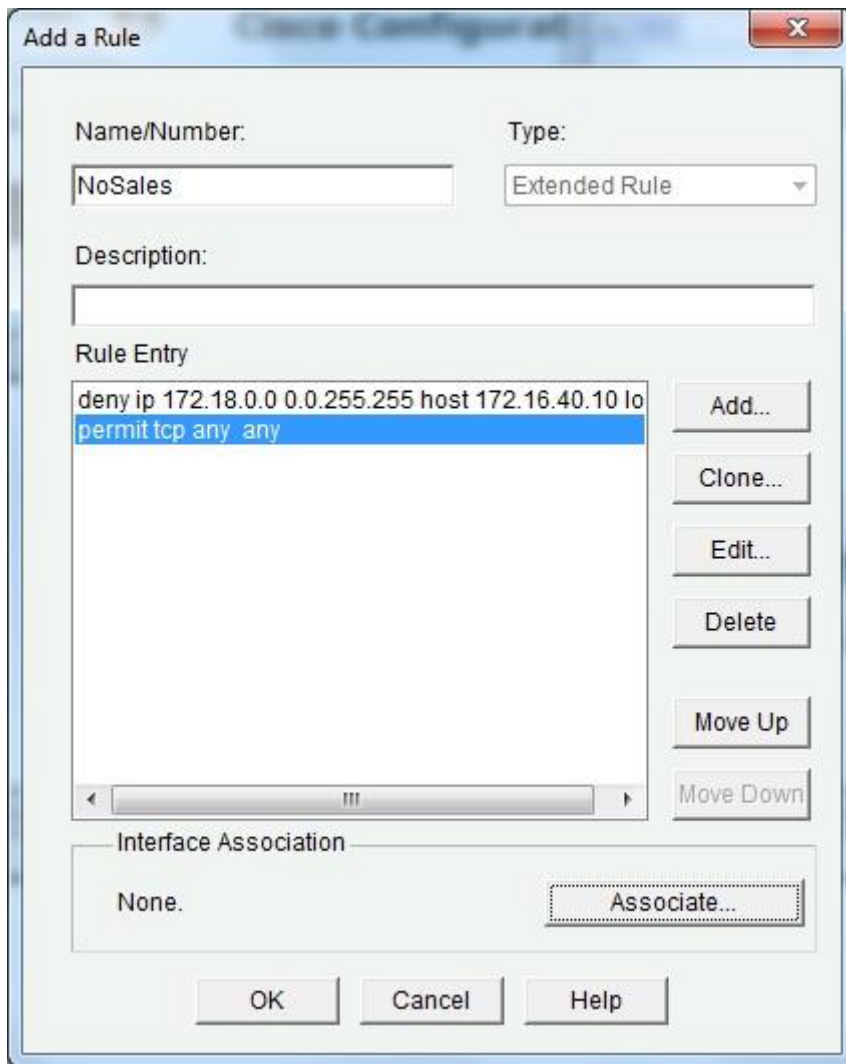
☐ Match established connections

☐ Log matches against this entry

OK Cancel Help

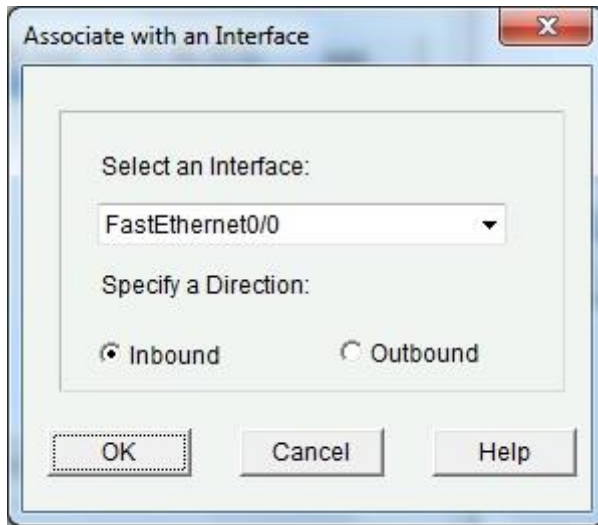
The access list has been created by now as shown in Figure 9-17, and we need to apply it to an interface. Press *Associate* to proceed.

**Figure 9-17** Dialogue – Add a Rule



We apply the access list to interface FastEthernet0/0 in the inbound direction, as shown in Figure 9-18.

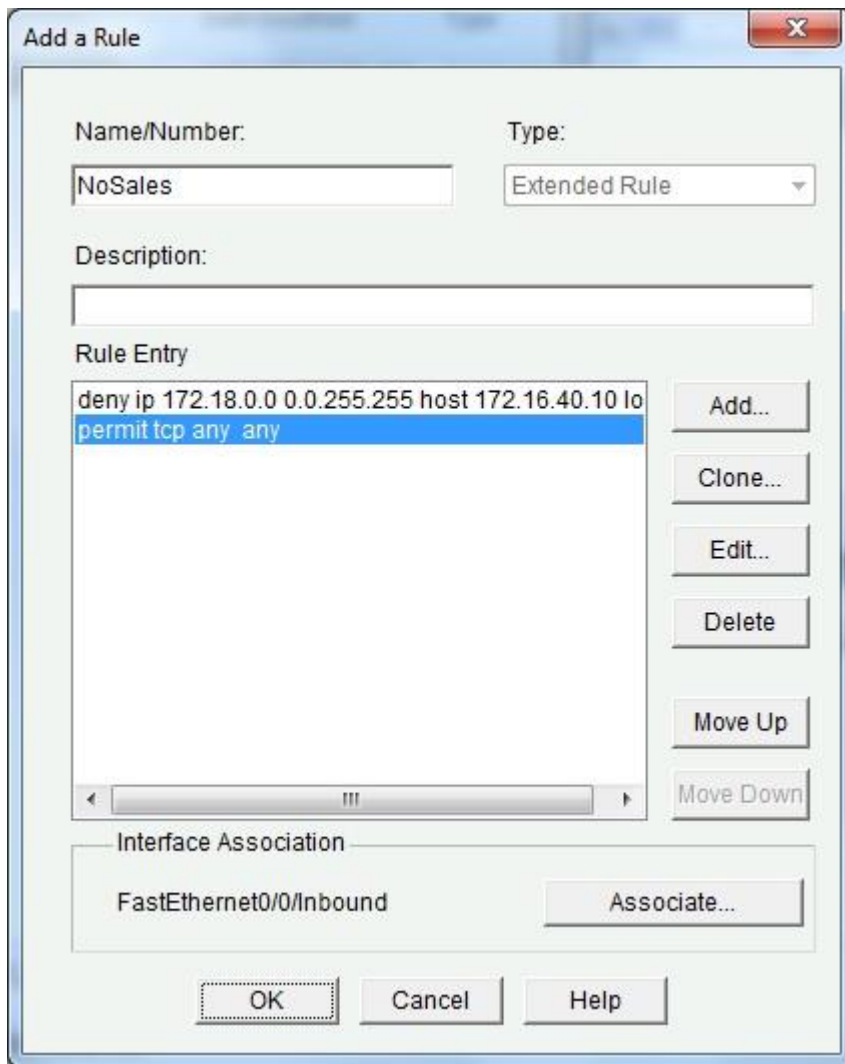
**Figure 9-18** Dialogue – Associate with an Interface



The configuration is complete and you return to the *Add a Rule* dialogue box, as shown in Figure 9-19. Simply press *OK* to proceed.

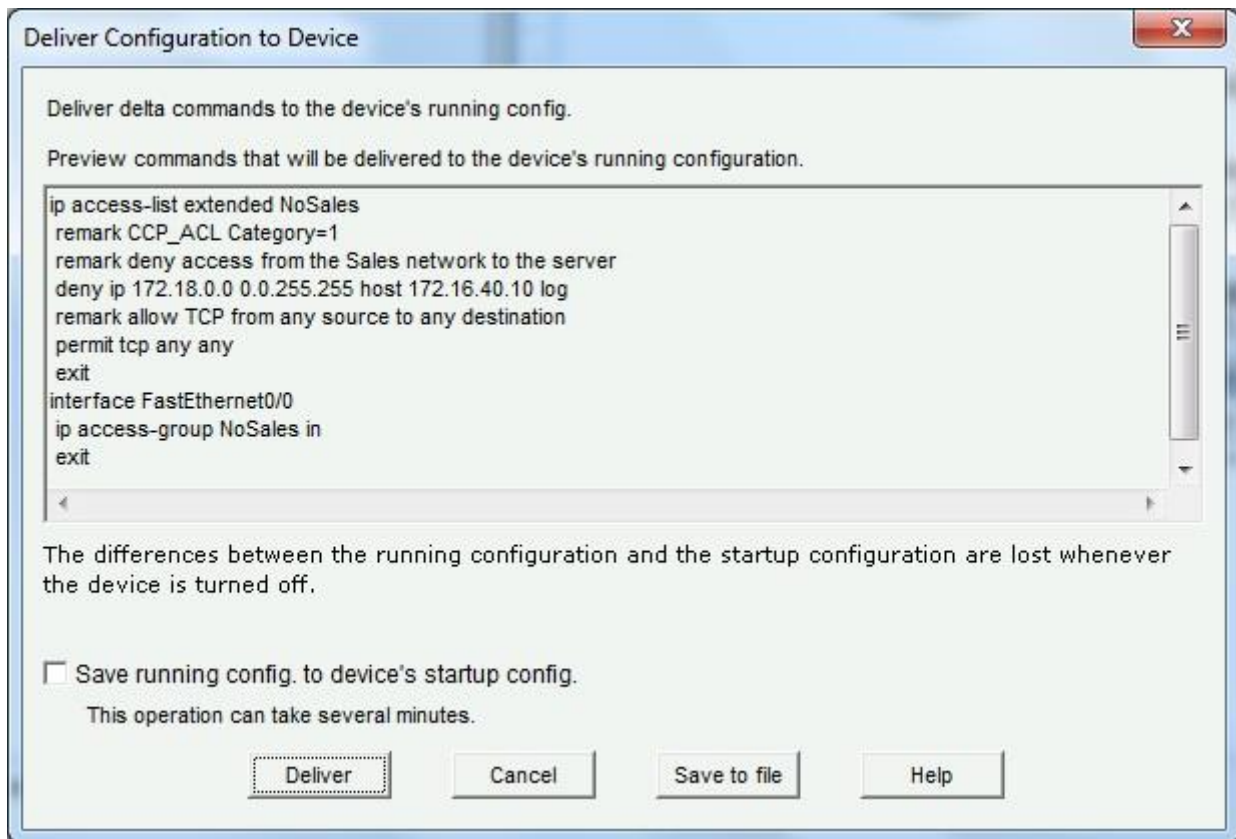
**Figure 9-19** Dialogue – Add a Rule





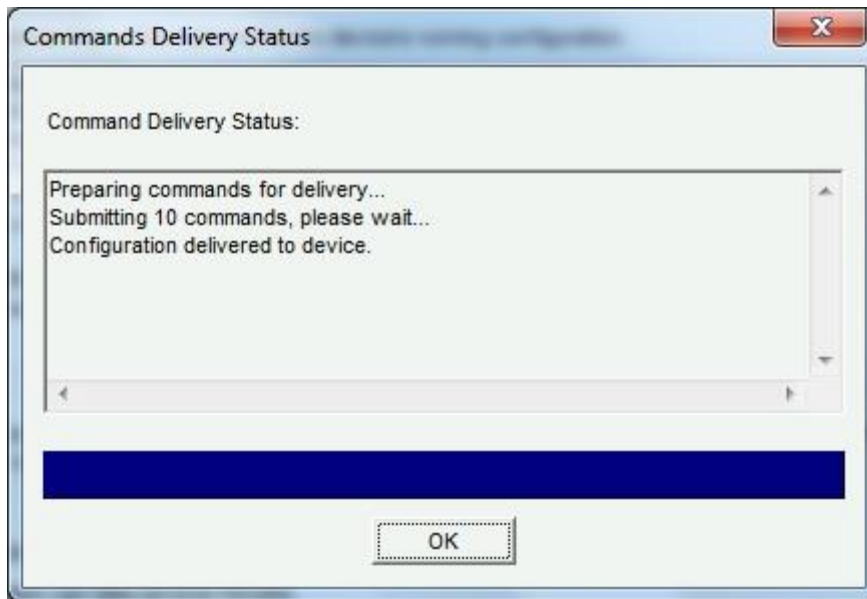
Another box appears that displays configuration that would actually be applied to the router, as shown in Figure 9-20. You can see that the configuration that actually gets applied to the router is just the same we created in an earlier section of the chapter. Press *Deliver* to apply the configuration to the running configuration. You may choose to select the *Save running config to device's startup config* checkbox to save the running configuration to startup configuration as well.

**Figure 9-20** Dialogue – Deliver Configuration to Device



Command delivery status looks good, as shown in Figure 9-20. We're done with creating an access list using Cisco Configuration Professional. You may press *OK* to return to the main application window.

**Figure 9-20** Command Delivery Status



Once you have come this far setting up Cisco Configuration Professional, it is a good idea to explore the configuration options available. It would be fun and a great way to learn Cisco CP while doing so. You are sure to get amazed with what you can do with Cisco CP with minimal knowledge of the Cisco CLI.

## Summary

We dedicated this chapter almost exclusively to access control lists (ACLs). Access lists are generally used for traffic filtering but they are quite versatile and have several other uses as well that were briefly mentioned in the beginning of the chapter

We covered both standard and extended access lists in detail and learned how to configure them in both named and numbered formats. Several nuances of access lists were also covered from a practical standpoint.

The chapter concluded with coverage of Cisco Configuration Professional (Cisco CP), a GUI based tool that can be used to configure and manage Cisco devices.

## **Chapter 10 – Network Address Translation (NAT)**

In this chapter, we are going to learn Network Address Translation (NAT) as configured on Cisco routers in common network scenarios. First time NAT users find NAT concepts difficult to grasp and NAT configuration difficult to create and understand. But the fact of the matter is that the basic concepts of NAT are pretty simple as we will see below.

After developing an intimate understanding of basic NAT concepts early in the chapter, we will learn how to configure and troubleshoot NAT using the Cisco command line interface (CLI). Toward the end of the chapter, we will also see how NAT can be configured the easy way using Cisco Configuration Professional (Cisco CP).

- 10-1 Introduction to NAT
- 10-2 Static NAT Configuration & Verification
- 10-3 Dynamic NAT Configuration
- 10-4 NAT Overloading aka Port Address Translation (PAT)
- 10-5 NAT Troubleshooting
- 10-6 NAT Configuration with Cisco Configuration Professional

## **10-1 Introduction to NAT**

### **What is NAT?**

Network Address Translation (NAT) allows a host that does not have a registered IP address to communicate with other hosts on the Internet. NAT has gained such widespread acceptance that the majority of enterprise networks today use private IP addresses for most hosts on their network and use a small block of public IP addresses, with NAT translating between the two.

Having come this far in your CCNA studies, you should be well aware of the IP header format. The IP packet header has several fields in it, the most well-known of which probably are the *Source IP Address* and *Destination IP Address*. NAT simply translates, or changes, one or both of these addresses inside a packet header as the packet passes through the router performing the NAT operation. This is what basic NAT operation is, nothing more, nothing less.

### **Purpose of NAT?**

NAT is a feature that allows the internal network of an organization to *appear* to be using a different IP address space from the outside than what it is actually using. Thus, NAT allows an organization to use private IP addresses that are not globally routable and yet connect to the Internet by translating those private addresses into globally routable addresses. The beauty of NAT is that the hosts on the internal network using NAT to communicate to the outside world don't have to be aware of the very existence of NAT. NAT configuration exists only on the router or another device typically at the boundary of the internal network. Due to this aspect of NAT, an organization can also change service providers without any changes to the IP addresses configured on individual hosts. Changing service providers also changes public IP addresses available to an enterprise. The device performing NAT can have its configuration modified and that's all you need to do while changing your ISP.

### **Benefits of NAT**

Internet Protocol (IP) or IPv4 as it is more precisely known uses addresses that are 32-bits long. As such the total address space of IP is from 0 to  $2^{32} - 1 = 4,294,967,295$ . In other words, over four billion unique IP addresses are available for assignment to hosts. These IP addresses are the registered IP addresses that are centrally administered. Though four billion may seem like a very large number, due to the explosive growth of the Internet over the years we have already depleted most of those IP addresses.

RFC 1918 specifies three blocks of IP address space reserved by Internet Assigned Numbers Authority (IANA) for private networks.

**Table 10-1** Private IP Addresses

Address Class	Number of Networks	Private Address Space
A	1	10.0.0.0 – 10.255.255.255
B	16	172.16.0.0 – 172.31.255.255
C	256	192.168.0.0 – 192.168.255.255

An enterprise can assign these private IP addresses to internal host without the need for registered IP addresses. A router or other device can be used to perform Network Address Translation (NAT) to convert these private IP addresses to public IP addresses routable on the Internet.

Network Address Translation (NAT) is defined in RFC 1631. The original intention of NAT was to slow the depletion of available IP addresses by allowing many private IP addresses to be represented by some smaller number of public IP addresses. NAT was envisioned to be a temporary solution to the problem of IP version 4 address depletion. The permanent solution was a migration from IP version 4 to IP version 6 (IPv6). IPv6 has 128-bit addresses and a much larger address space expected to solve the issue of address scarcity forever. But NAT has been so successful that it has delayed the full IPv4 address exhaustion by several years.

NAT has also come to find other applications that do not directly relate to IP address conservation. One such NAT application is the merger of two companies and hence their internetworks. The two companies would previously have two separate internetworks. After the merger their two internetworks would have to be connected together. Unfortunately, when the two separate internetworks were first constructed several years ago, nobody had anticipated a future merger. So the designers of both internetworks chose to use the 10.0.0.0 address space. As a result, many IP addresses would be assigned to devices in both internetworks. NAT can be used as a temporary solution to connect the internetworks. Keep in mind that the best solution in such a situation is to re-address the new internetwork. But re-addressing can be a major project if all the devices have manually configured IP addresses. NAT can serve as an interim solution.

NAT allows organizations to solve the problem of IP address depletion when they want to connect new networks to the Internet. NAT allows organizations to connect their networks to the Internet without needing to have Network Information Center (NIC) registered IP addresses assigned to their internal systems.

There are organizations that already have registered IP addresses for hosts on an internal network but they want to hide those addresses from the Internet so that hackers cannot easily attack those hosts. If the host address is hidden, a degree of security is achieved. NAT can be useful in this situation where the motive is not primarily IP address preservation, but applying corporate security policies to your network traffic.

A major advantage of NAT is that it needs to be configured only on those few routers that would actually perform the NAT operation. The hosts or other routers not performing NAT operation don't need any configuration changes.

### **Disadvantages of NAT**

Network Address Translation (NAT) is all about changing IP addresses and port numbers inside an IP packet header which also creates some issues. Changing the

content of an IP address or TCP port can change the meaning of some of the other fields, especially the checksum. For example, the checksum of an IP packet is calculated over the entire IP header. Therefore if the source or destination IP address (or both) change, the checksum has to be calculated again. The same is also true for the checksum in the TCP header. This number is calculated over the TCP header and data, and also over a pseudo-header that includes the source and destination IP addresses. Therefore, if an IP address or a port number changes, the TCP checksum must also change. NAT as implemented on Cisco routers performs these recalculations. This is extra work for the router performing NAT.

NAT should be transparent to the end systems that send packets through it. However, many applications use the IP addresses at the application layer. Information within the data field may be based on an IP address, or an IP address itself may be carried in the data field. If NAT translates an address in the IP header without being aware of the effects on the data, the application breaks.

As a matter of fact, Cisco's NAT implementation goes beyond translating addresses in the IP header for the applications it supports. For the supported applications carrying IP address information in the application data, NAT makes the appropriate corrections to the data as well. This prevents the application from breaking due to NAT.

However, if the data fields are encrypted, NAT has no way of reading the data. Therefore, for NAT to function properly, neither the IP addresses nor any information derived from them (such as the TCP header checksum) can be encrypted. But this is not the case with virtual private networks (VPNs), for example, IPsec. With certain modes of IPsec, if an IP address is changed in an IPsec packet, the IPsec becomes meaningless and the VPN is broken. When any sort of encryption is used, you must perform NAT on the secure side before encryption, rather than in the encrypted path.

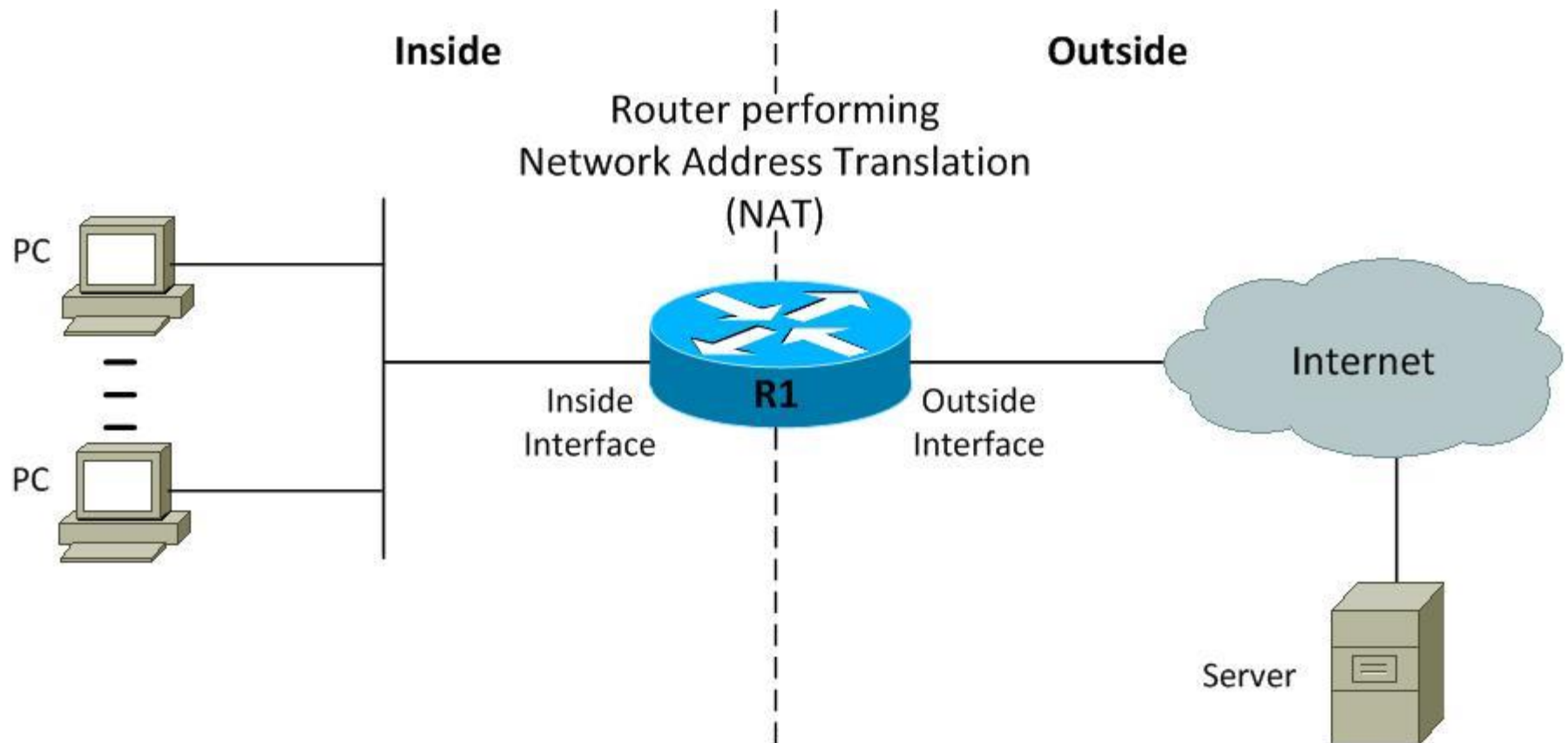
NAT is also viewed sometimes as part of a security plan, because it hides the details of the inside network from the outside world. A host with translated address may appear on the Internet with one address one day and with a different address on another day. But keep in mind that this is very weak security at best. It might slow down an attacker who wants to hit a particular host but it will not stop him if he is determined.



## NAT Inside and Outside Addresses

Let's define a few basic but important terms in the context of NAT. Before we jump into the definitions, have a look at Figure 10-1 in order to understand the context in which NAT typically operates.

**Figure 10-1** NAT Context



A device performing Network Address Translation (NAT) divides its universe into the *inside* and the *outside*. Typically the inside is a private enterprise with its internal network and hosts connected to that network. The outside on the other hand is the public Internet and the servers reachable over it. In addition to the notion of inside and outside, a Cisco NAT router classifies addresses as either *local* or *global*. A local address is an address that is seen by devices on the inside, and a global address is an address that is seen by devices on the outside.

Given these four terms, an address may be one of four types:

1. **1. Inside local** addresses are assigned to inside devices. These addresses are not advertised to the outside.
2. **2. Inside global** are addresses by which inside devices are known to the outside.
3. **3. Outside local** are addresses by which outside devices are known to the inside.
4. **4. Outside global** addresses are assigned to outside devices. These addresses are not advertised to the inside.

## Types of NAT

In general, NAT is configured on a Cisco router that connects only two networks, and translates the inside local (private) addresses from the internal network into inside global (public) addresses. In most common scenarios the outside addresses are not translated so outside global and outside local addresses are the same. You can configure NAT in a way that it will advertise only a single address for your entire network to the outside world. Doing this effectively hides the addresses in your internal network from the hostile environment of the Internet. Thus, giving you some additional security and peace of mind as network administrator.

NAT has the following types:

- **Static NAT:** Static NAT performs static address translation allowing one-to-one mapping between local and global addresses. But you should keep in mind that static NAT requires you to have one registered public IP address for every host on your network. As such static NAT has no benefit in terms of IP address conservation. Nevertheless, static NAT is important for the sake of understanding NAT.
- **Dynamic NAT:** Dynamic NAT performs dynamic address translation mapping unregistered private IP addresses to registered public IP addresses from a pool of available registered IP addresses. You don't have to statically configure your router to map an inside to an outside address as you would using static NAT. But yet you do have to have enough registered public IP addresses for everyone who's going to communicate to the Internet. Even dynamic NAT does not help with the issue of IP address conservation.
- **NAT Overload:** NAT overload performs an overload mapping multiple unregistered private IP addresses to a single registered public IP address. It is a many-to-one mapping between private and public addresses and is accomplished using different port numbers. This method is also known as Port Address Translation (PAT). By using PAT or NAT overload, hundreds or even thousands of users can be connected to the Internet using only one real global IP address. This is the most popular NAT type which basically is a form of dynamic map but with multiple unregistered IP addresses mapped to a single registered IP address. Dynamic NAT is one-to-one while NAT Overload or PAT is many-to-one though both forms do the mapping dynamically. NAT Overload is the type of NAT that has enabled us not to run out of IP addresses on the Internet.



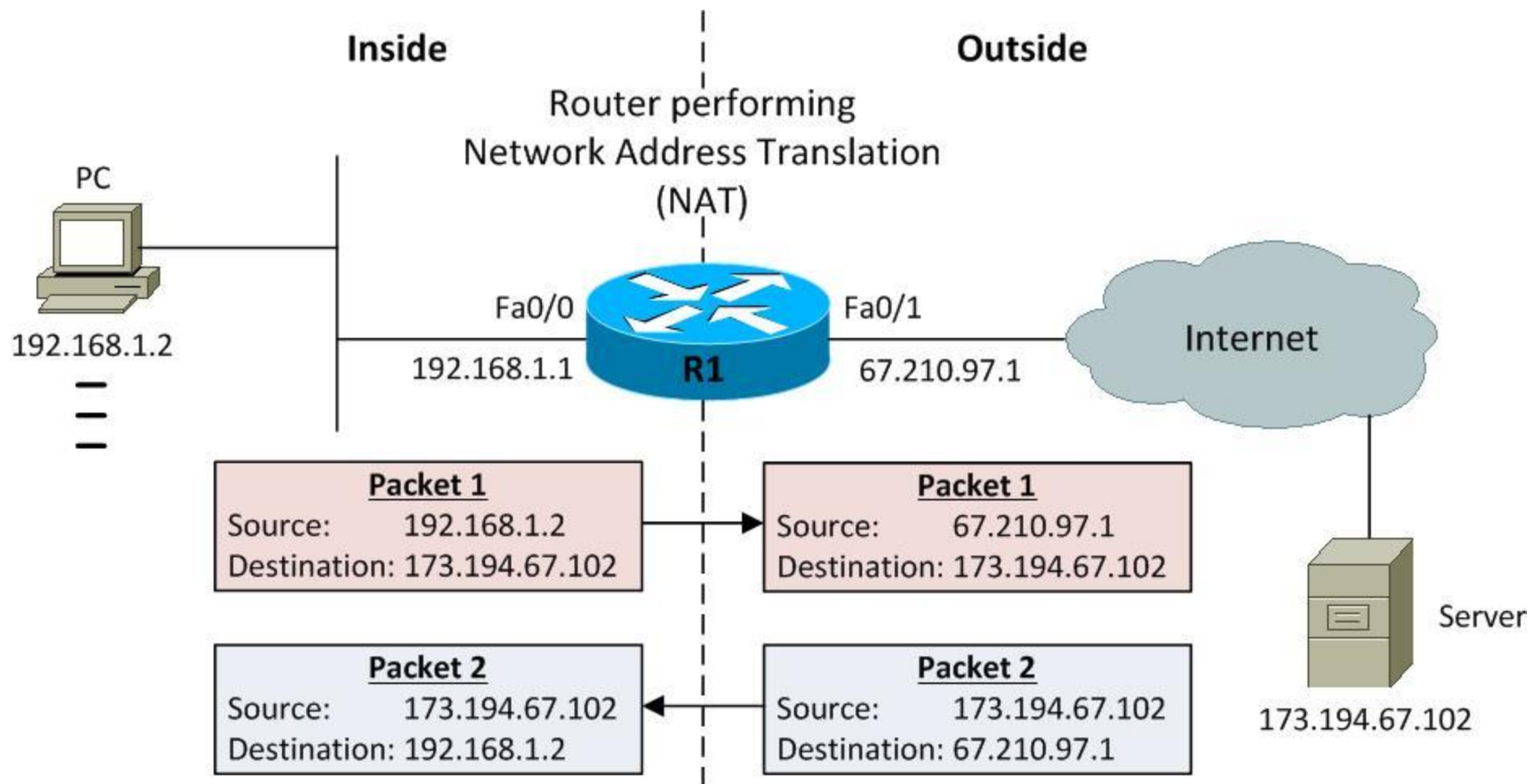
**Exam Concept –** The three methods of Network Address Translation (NAT) are static, dynamic, and overloading which is also called Port Address Translation (PAT). You are sure to see questions on the CCNA exam contrasting each.

We will learn how to configure each of these three types of NAT in this chapter.

### How NAT Operates?

Let's have a look at Figure 10-2 that present a basic NAT scenario. We will explain the basics of NAT operation with the help of this scenario and the understanding you develop here should also help you understand the rest of the chapter.

**Figure 10-2** Basic NAT Operation



We have defined several terms related to NAT so far. It's the right time to test that understanding in the context of Figure 10-2. R1 is the router performing NAT and has two interfaces: Fa0/0 as the inside interface while Fa0/1 is the outside interface. A PC having IP address 192.168.1.2 on the inside needs to communicate with a server having IP address 173.194.67.102 on the outside.

**Table 10-2** NAT Address Types

<b>NAT Address Type</b>	<b>IP Address</b>
Inside local	192.168.1.2
Inside global	67.210.97.212
Outside local	173.194.67.102
Outside global	173.194.67.102

Let's try to understand what happens to IP packets travelling back and forth between the PC and the server as they move across R1 – the router performing NAT operation. In plain simple words IP addresses in the header of those IP packets get re-written with R1 also keeping a record of these re-writes or translations in a table known not surprisingly as translation table.

Let's first consider a packet that moves from inside to outside. This packet has source and destination IP addresses of 192.168.1.2 and 173.194.67.102 respectively. The packet enters router R1 at its inside interface Fa0/0 and exits the outside interface Fa0/1. Before the packet actually exits R1, the source address gets re-written. The inside local IP address 192.168.1.2 is replaced with the inside global IP address 67.210.97.212. The destination IP address is left untouched here so the outside local and outside global address are both 173.194.67.102. And some of you may have identified that the inside global address is in fact the IP address configured on the outside interface Fa0/1 of R1. To the outside world, the packet appears to have originated from the IP address 67.210.97.212 and the inside local IP address 192.168.1.2 is never known to the outside world! For packets moving in the opposite direction, from outside to inside, the destination IP address 67.210.97.212 gets re-written with 192.168.1.2 while the source IP address remains unchanged. The PC and

server can communicate successfully yet the server or any entity on the outside does not know the real IP address of the PC.

In the coming sections we would learn in detail how to actually configure Network Address (NAT) on a Cisco router for all three types of NAT.

## **10-2 Static NAT Configuration & Verification**

With static NAT, a particular inside local address always maps to a particular inside global (public) address. Due to one-to-one mapping between addresses static NAT does *not* conserve public IP addresses. Although static NAT does not help with IP address conservation, it provides a degree of security by hiding the inside IP addresses from the outside world. Static NAT also allows an administrator to make an inside server available to clients on the Internet, because the inside server will always use the same public IP addresses.



**Exam Concept** – NAT mapping commands are used in global configuration mode. It is common for Cisco to provide you the correct command at different modes on the CCNA exam. Thus know the mode in which the command is executed.

Let's configure router R1 performing NAT as depicted in Figure 10-3.



**Figure 10-3** NAT Scenario

Here is how the configuration goes.

The **ip nat inside source** command identifies which IP addresses will be translated. In the preceding configuration example, the **ip nat inside source** command configures a static translation between inside local and inside global IP addresses as shown in Table 10-2 below.

**Table 10-3** Static NAT Address Mapping

Inside Local Addresses	Inside Global Addresses
192.168.1.2	67.210.97.2
192.168.1.3	67.210.97.3
192.168.1.4	67.210.97.4

You may also identify an **ip nat** command under each interface in the above configuration. The **ip nat inside** command identifies an interface as the inside interface.

The **ip nat outside** command identifies an interface as the outside interface. The **ip nat inside source** command is actually referencing the inside interface with the **inside** keyword and the **source** address with the source keyword.

The **static** keyword indicates a static one-to-one mapping between inside local and inside global addresses.



**Exam Concept** – Static NAT is designed to allow one-to-one mapping between local and global addresses.

The **ip nat inside source** command is simply instructing the router to translate the source address of every packet entering the router at the inside interface. In order to ensure two-way communication return packets coming in the outside interfaces are also translated accordingly.

Once you finish your NAT configuration, you would usually want to verify if the configuration is working as expected or not. Also, you may need to monitor NAT translations in a production environment. It is quite tempting to use **show running-config** command to verify that the NAT configuration lines you entered are actually there in the running configuration of the router. But this does not tell you anything about whether actual translation of addresses is taking place or not.

The starting point for NAT verification and troubleshooting should be the **show ip nat translations** command:

The output above shows inside local addresses mapped to inside global address as configured on R1 and summarized in Table 10-2. What's shown in the above output

does not include any real translations so far because inside hosts did not send any traffic yet. What we see so far is the result of the three **ip nat inside source** commands in our configuration. In order to see some real translations we would generate some traffic by pinging the server 173.194.67.102 from each of the three inside hosts and run the show ip nat translations command again on R1:

```
R1#show ip nat translations
Pro  Inside global  Inside local  Outside local  Outside global
icmp 67.210.91.2:6  192.168.1.2:6  173.194.67.102:6  173.194.67.102:6
--- 67.210.91.2    192.168.1.2    ---              ---
icmp 67.210.91.3:7  192.168.1.3:7  173.194.67.102:7  173.194.67.102:7
--- 67.210.91.3    192.168.1.3    ---              ---
icmp 67.210.91.4:8  192.168.1.4:8  173.194.67.102:8  173.194.67.102:8
--- 67.210.91.4    192.168.1.4    ---              ---
```

In the above output, you can see three dynamic entries show in gray, each corresponding to a ping from an inside host to the server 173.194.67.102. Typically you would see several dynamic entries in the translation table for the same inside host communicating with several outside hosts using various protocols.

Another useful tool for NAT verification is the **show ip nat statistics** command:

This command provides NAT statistics including the number of translated packets or hits.

Please note that translation table entries eventually time out. However you may also use the **clear ip nat translations** command to clear specific entries from the translation table before they time out on their own. In order to clear all entries from the translation table, use an asterisk (\*) at the end of the command:

As you can see in the above output, the ICMP translation entries are all gone following the **clear ip nat translations \*** command.

### **10-3 Dynamic NAT Configuration**

Dynamic NAT also creates one-to-one mappings between addresses and does not conserve IP addresses, just like static NAT. However, dynamic NAT creates a pool of inside global IP addresses to be mapped to an access list identifying inside local IP addresses. So basically we have two sets of addresses being mapped and not individual addresses. The same inside local address may not map to the same inside global address every time. The configuration should help make these concepts more understandable.

Here is a sample dynamic NAT configuration for the scenario in Figure 10-3.

There are three parts of the above configuration.

First, the command **ip nat pool MyPool 67.210.97.2 67.210.97.4 netmask 255.255.255.0** is used to create a pool of inside global addresses from 67.210.97.2 to 67.210.97.4. That is a total of 3 addresses only with a subnet mask of 255.255.255.0. Please note that we chose **MyPool** as NAT pool name but this choice is arbitrary and NAT pool name can be anything you like, even your first name. Also note that a network mask has to be specified using **netmask** keyword when defining a NAT pool.

Second, the **ip access-list 1** commands create a standard access list matching interesting traffic for address translation. The access list would match IP addresses of the three inside hosts.

Third and last, the **ip nat inside source list 1 pool MyPool** command instructs the router to dynamically translate source IP addresses of packets coming in at the inside

interface that match **access-list 1** to an address found in the **ip nat pool** named **MyPool**.



**Exam Concept** – Dynamic NAT allows one-to-one mapping of local addresses to global addresses from a pool of global addresses.

Let's verify it now:

There is no output so far as there are no static mappings between inside local and inside global addresses. Let's generate some traffic from each of the three inside hosts and run the **show ip nat translations** command again:

Let's issue the **clear ip nat translations \*** command and view the translation table again:

The translation table is empty now as there were no entries as a result of static mapping.

If you can recall what we learned in the chapter on access lists, access lists were presented as tools to match packets comprising of *interesting* traffic. The access lists here is also being used to match interesting traffic for address translation. The access list is not used for traffic filtering because the access list was never applied to an interface using **ip access-group** command.

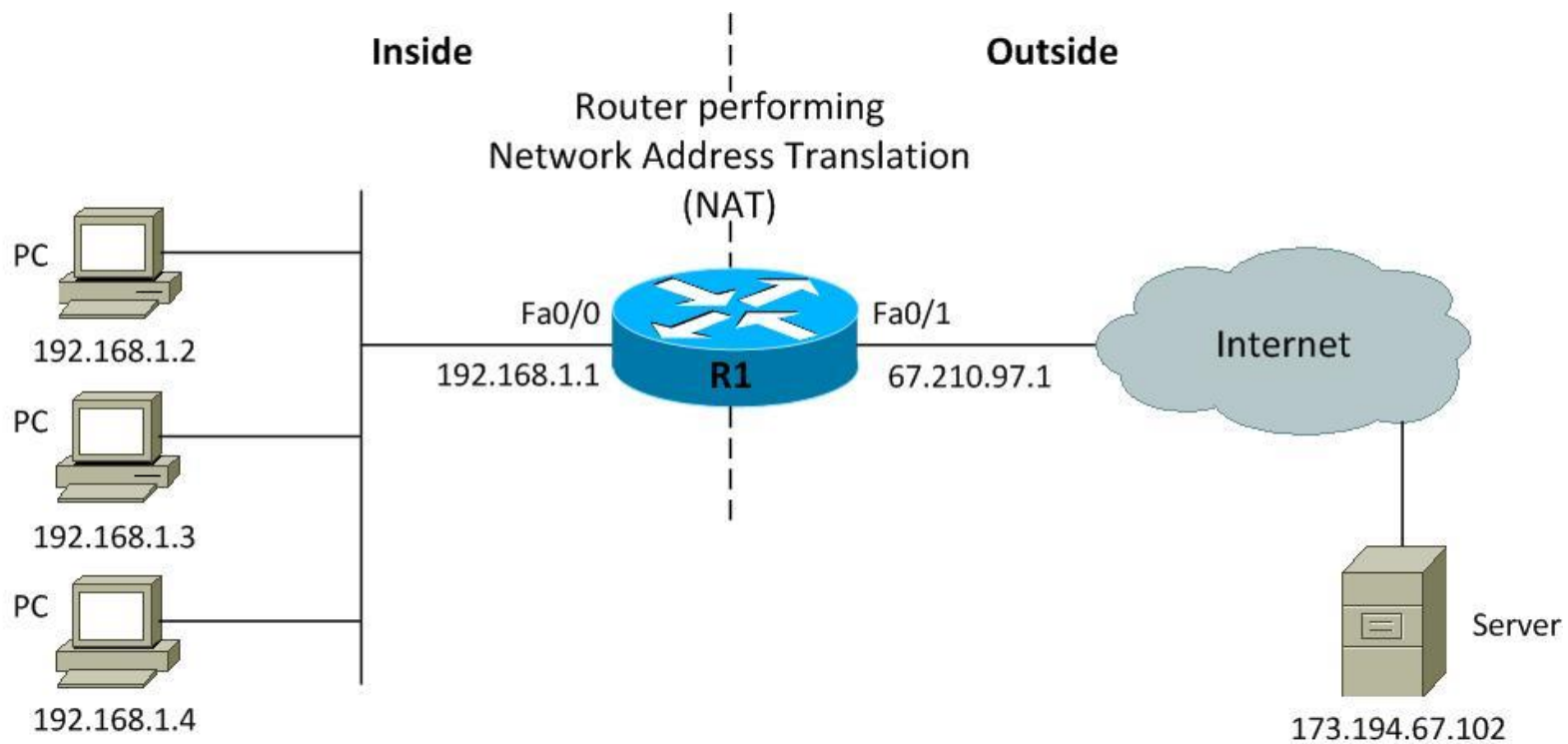
Please keep in mind that both static and dynamic NAT create one-to-one mapping of inside local and inside global addresses. The only difference is that for static NAT we need to specify explicitly which inside local address maps to which inside global address. While, for dynamic NAT we just have to create an access list to identify inside local addresses and a pool to specify inside global addresses. The actual mapping is done dynamically as the router performing NAT receives interested packets.

### **10-4 NAT Overloading aka Port Address Translation (PAT)**

NAT Overloading or Port Address Translation (PAT) is a modified form of dynamic NAT where the number of inside local addresses is greater than the number of inside global addresses. Mostly, there is just a single inside global IP address providing Internet access to all inside hosts. NAT Overloading is the only flavor of NAT that actually conserves IP addresses and it is also the most popular form of NAT as well.

Figure 10-4 Port Address Translation (PAT)





**NAT Translation Table**

Protocol	Inside Local IP : Port	Inside Global IP : Port
ICMP	192.168.1.2 : 18	67.210.97.1 : 18
ICMP	192.168.1.3 : 19	67.210.97.1 : 19
ICMP	192.168.1.4 : 20	67.210.97.1 : 20

PAT allows overloading or the mapping of more than one inside local address to the same inside global address. But this also means that return packets would all have the same destination address as they reach the NAT router. How would the router know which inside local address each return packet belongs to? In order to deal with this scenario, the NAT entries in the translation table are extended entries; the entries not only track the relevant IP addresses, but also the protocol types and ports. By translating both the IP address and the port number of a packet, up to 65535 inside local addresses could theoretically be mapped to a single inside global address (based on the 16-bit port number).

But keep in mind that a single NAT entry uses approximately 160 bytes of router memory, so 65535 entries would take more than 10 MB of memory and also large amounts of CPU power. In practical PAT configurations, nowhere near this number of addresses are mapped, but it is definitely a theoretical limit.

Here is a sample configuration for NAT overloading or PAT according to Figure 10-4.

The above configuration may appear very similar to the configuration for dynamic NAT, however there are important differences. First, the pool of IP addresses has been shrunk to a single IP address assigned to the outside interface of router R1.

Second, access list 1 matches the entire class C network 192.168.1.0/24 which means any inside local address from this network will be translated. If you want a specific host from this network not to be translated, you have to explicitly specify by adding a deny statement to the access list. Let's assume we want to deny translation to a single host 192.168.1.2 while allowing all other hosts:

Also there is an addition of **overload** keyword with the **ip nat inside source list 1 pool MyPool** command.



**Key Concept** – NAT Overload is a special form of dynamic NAT that allows many-to-one mapping of local addresses to a smaller number global addresses from a pool of global addresses. The pool of global addresses may even consist of a single address. NAT Overload is also called Port Address Translation (PAT). These are a favorite type of scenario question on the CCNA exam.

Let's start our usual verification by issuing the **show ip nat translations** command:

There are no static mappings and hence the blank output above. Let's generate some traffic from inside hosts to the server and issue the **show ip nat translations** command again:

As you can see in the output above, all inside local addresses are translated into the same inside global address, which is the essence of NAT overload or PAT. You may have noticed that the router has preserved the source port numbers as inside local addresses were translated to inside global addresses. This is the usual behavior but when the router creates a new translation entry such that the source port number is already in use, the port number also gets translated to a different number. The occurrence of two inside hosts choosing the same source port number is not very common, but it still may happen especially when the number of connections from inside to outside is significant.

Let's also issue the **show ip nat statistics** command:

You should keep two things in mind. First, NAT overload is useful in any situation when the number of inside hosts is larger than public addresses you have. In many situations you only have a single public IP address that is assigned to the outside interface of your Internet facing router. In this case your pool would consist of a single IP address as the configuration above shows. However, you may have more than one public IP addresses available, one of which may be assigned to the Internet facing interface of your router. In such case, your NAT pool may consist of more than one IP addresses still using NAT overload to accommodate a larger number of inside hosts wanting to connect to the Internet. In short, you may have a single overloaded public address or you may have more than one overloaded public addresses.

NAT Overload or PAT is the most prevalent NAT configuration for the obvious reason that it is the flavor of NAT that actually preserves global IP addresses, the primary reason for NAT usage.

## **10-5 NAT Troubleshooting**

NAT mechanics are complex, but NAT configuration is pretty simple. However, if something does not work as expected, there are a bunch of things you can do. The **show ip nat translations** and **show ip nat statistics** commands covered in previous sections usually provide enough information to be able to identify problems with NAT. However there is another useful tool you should probably have in your toolbox and that is the **debug ip nat** command:

The above output is from router R1 configured with Static NAT as presented earlier in the chapter. Nothing is broken here and the configuration is good. We generate a single ping from each of the inside hosts 192.168.1.2, 192.168.1.3, 192.168.1.4 to the server 173.194.67.102. You can see six debug entries in the above output for outgoing as well as return packets. In the outgoing packets, the source IP address is translated. While in the return packets, the destination IP address is translated.

The **debug ip nat** command can be used to verify the operation of NAT displaying information about each packet the router translates. This command also displays information about certain errors, such as the failure to allocate a global address.

As a general rule, you should always use **show** commands first for verification and troubleshooting. All **debug** commands should be used only when you have exhausted your options with **show** commands. These **debug** commands consume resources namely CPU cycles and memory, and should be used with caution on production networks especially if you love your current job.

## **10-6 NAT Configuration with Cisco Configuration Professional**

Cisco Configuration Professional can be used for configuration of NAT on a Cisco router. If you have read Chapter 9, you should be familiar with the process of setting up Cisco CP. If not, it is probably the right time to go back to Chapter 9 and review that material before proceeding. We will configure NAT overloading or PAT using Cisco CP. PAT is the most prevalent form of NAT, so it makes sense to use it for this section on Cisco CP.

First, launch Cisco Configuration Professional to connect to the router R1 on which we want to configure NAT. We assume the device is already set up as shown in Chapter 9.

In the right pane select the device and press Configure button on the top left of the main application screen.

**Figure 10-5**



Select Community Member:

192.168.1.1

«

[Home > Community View](#)

 Cisco Configuration Professional News : Unavailable due to connection failure with [www.cisco.com](http://www.cisco.com)

### Community Information

Selected community: **New Community** . Select a device from the table below. Use the buttons at the bottom to continue.

Filter

| 1 rows retrieved |

[illegible]

### Manage Devices

Delete

Discover

### Discovery Details

Cancel Discovery

Router Status

## Utilities

- Flash File Management
- Configuration Editor
- Save Configuration to PC
- Write to Startup Configuration
- Telnet
- Reload Device

In the left pane, select *Router > NAT*. In the right pane, select the *Basic NAT* radio button and press the *Launch the selected task* button.

**Figure 10-6**

http://127.0.0.1:8600/Counterpoint/CPMain.html?rand=23964 - Windows Internet Explorer

ApplicationHelp

HomeConfigureMonitor

Cisco Configuration ProfessionalCISCO

Select Community Member:192.168.1.1<<Configure > Router > NAT

Interface Management

Interface and Connections

Router

Router Options

Time

Router Access

DHCP

DNS

Static and Dynamic Routing

AAA

ACL

NAT

QoS

Performance Routing

Router Provisioning

...

Utilities

Flash File Management

Configuration Editor

Save Configuration to PC

Write to Startup Configuration

Telnet

Reload Device

NAT

Create NAT ConfigurationEdit NAT Configuration

Cisco CP can guide you through NAT configuration tasks. NAT allows you to connect the hosts on your LAN to the Internet. Select a task, then click 'Launch the selected task' button.

Basic NAT

If you just have PCs or hosts on the LAN that need access to the Internet, select this option.

Advanced NAT

If you are hosting servers (e.g. web servers, e-mail servers) that users outside your network need access to, select this option.

Use Case Scenario

PC with private IP address

Internet

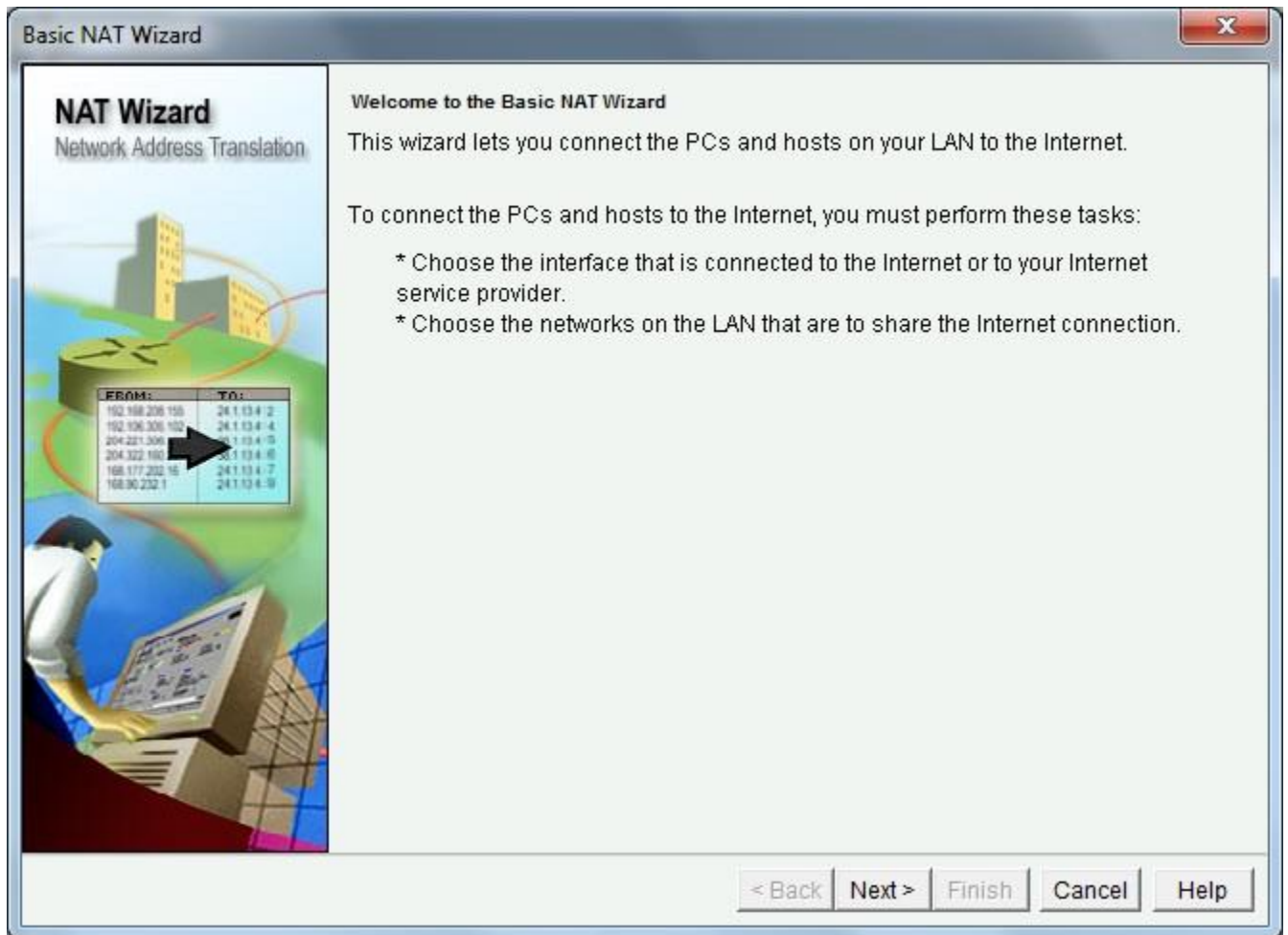
Launch the selected task

How do I:How do I Configure Address Translation for Outside to Inside?

Go

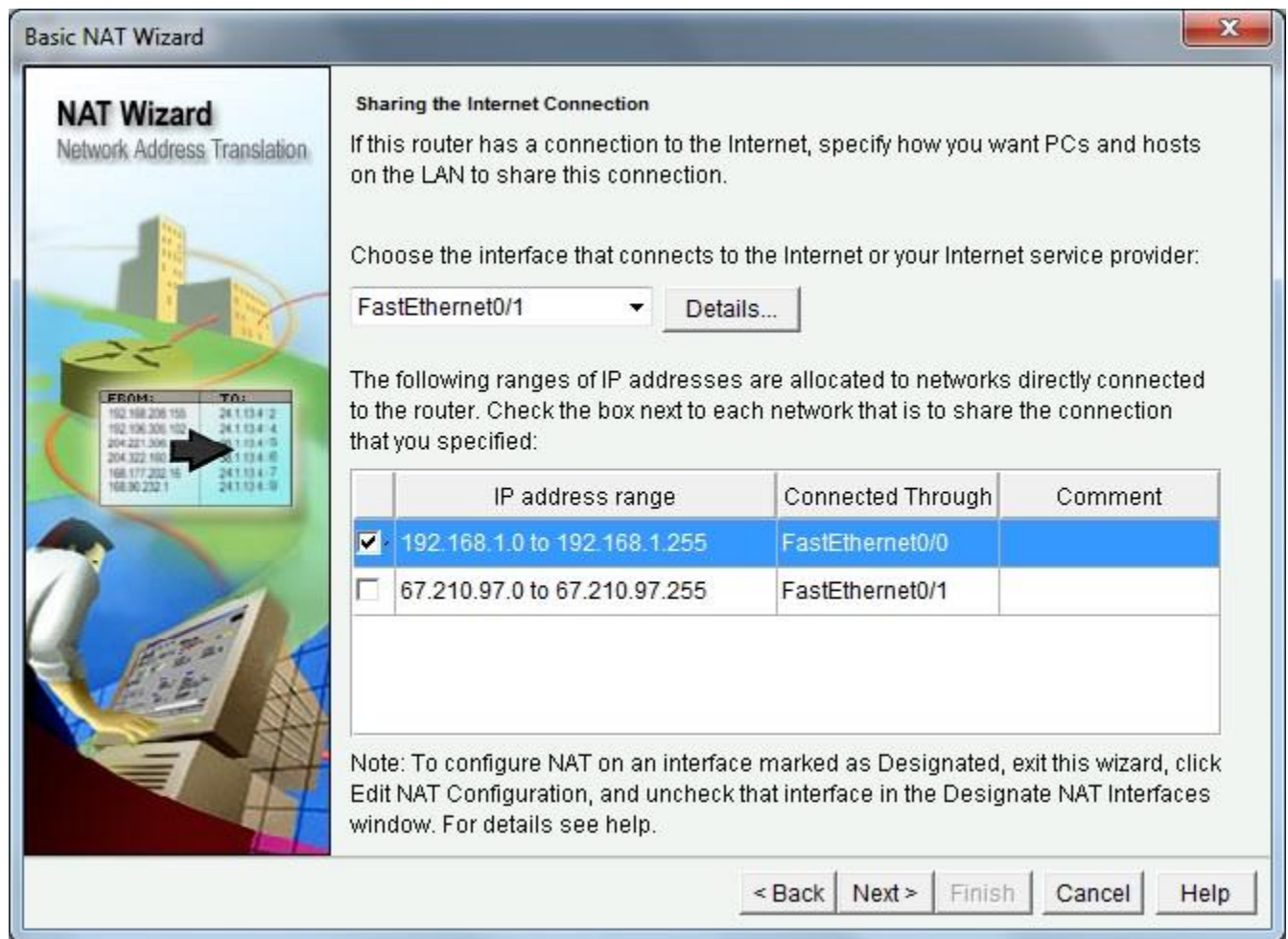
The Basic NAT wizard is launched and you just have to press the *Next* button to proceed.

Figure 10-7



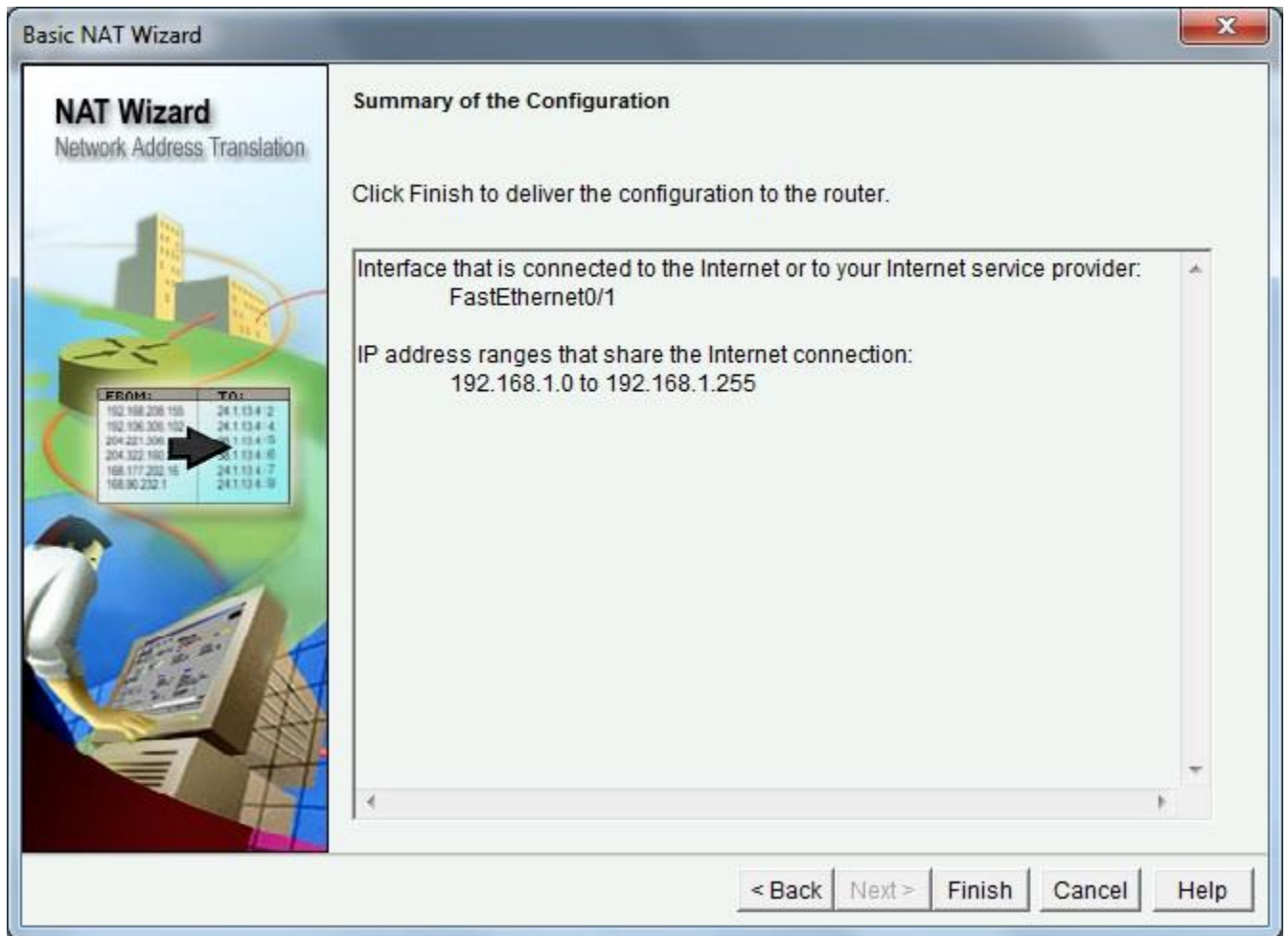
Choose FastEthernet0/1 as the interface that connects to the Internet from the drop-down menu. Also select the checkbox next to the FastEthernet0/0 network which is to share the connection to the Internet, and press the *Next* button to proceed.

Figure 10-8



Press the Finish button to proceed.

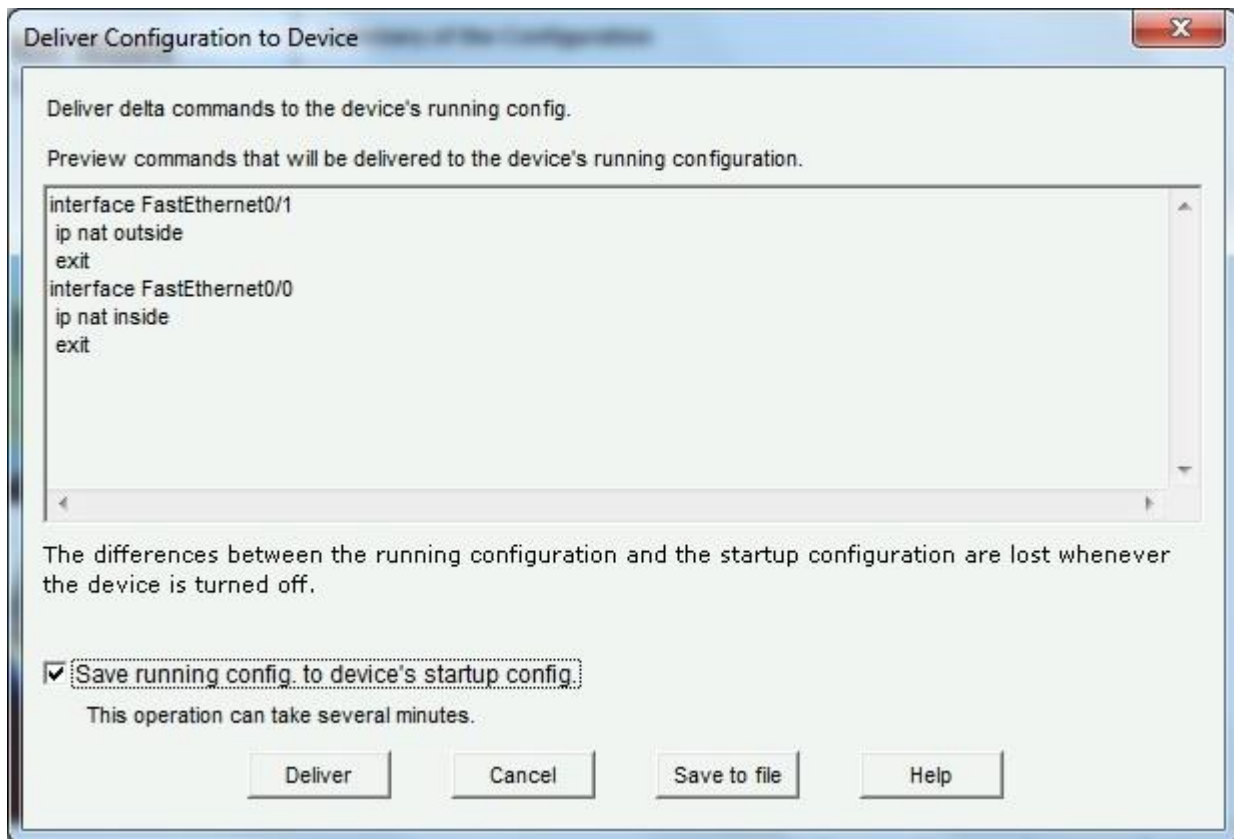
Figure 10-9



Select the *save running config. to device's startup config.* checkbox and press the Deliver button to send the configuration to the router and also to save it to the startup configuration.

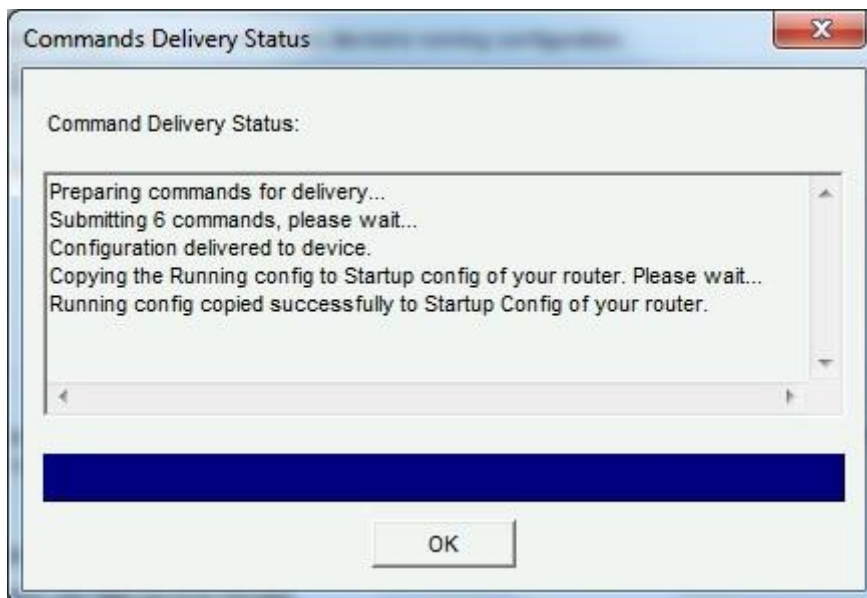
Figure 10-10





Press the OK button to proceed.

Figure 10-11



This finalizes your NAT Overload or PAT configuration using Cisco Configuration Professional.

**Figure 10-12**



http://127.0.0.1:8600/Counterpoint/CPMain.html?rand=23964 - Windows Internet Explorer

Application Help

Home Configure Monitor

Select Community Member: 192.168.1.1

Configure > Router > NAT

### NAT

Create NAT Configuration Edit NAT Configuration

Designate NAT Interfaces... Address Pool... Translation Timeouts...

Network Address Translation Rules

Inside Interface(s): FastEthernet0/0

Outside Interface(s): FastEthernet0/1

Original address	Translated address	Rule Type
192.168.1.0-192.168.1.255	67.210.97.1	Dynamic

Add... Edit... Delete View Route MAP...

☐ Clone selected Entry on Add

Utilities

- Flash File Management
- Configuration Editor
- Save Configuration to PC
- Write to Startup Configuration
- Telnet
- Reload Device

Cisco Configuration Professional CISCO

I would encourage you to connect router R1 and then using CLI examine the configuration manually paying special attention to NAT configuration. This is the configuration that was generated and delivered to the router by GUI wizards of Cisco CP.

## **Summary**

This is a really interesting chapter and you should have learned a lot about Network Address Translation (NAT) concepts and configuration. We configured NAT in three different flavors namely static, dynamic, and Port Address Translation (PAT) also known as NAT Overloading.

In order to stay focused on NAT concepts and not get lost in the intricacies of a complex topology, we used the same topology for different types of NAT configuration. It should have enabled you compare and contrast the three NAT types with relative ease.

We also went through some verification and troubleshooting commands before finishing the chapter by learning how to use Cisco Configuration Professional to configure NAT the quick and easy way.

## **Chapter 11 – Wide Area Networks**

- 11-1 Introduction to Wide-Area Networks
- 11-2 Point-to-Point WANs: Layer 1
- 11-3 Point-to-Point WANs: Layer 2
- 11-4 PPP Concepts
- 11-5 PPP Configuration
- 11-6 Troubleshooting Serial Links
- 11-7 Frame Relay
- 11-8 LMI and Encapsulation Types
- 11-9 Frame Relay Congestion Control
- 11-10 Frame Relay Encapsulation
- 11-11 Frame Relay Addressing
- 11-12 Frame-Relay Topology Approaches
- 11-13 Frame Relay Configuration
- 11-14 Other WAN Technologies

## **11-1 Introduction to Wide-Area Networks**

A wide-area network (WAN) enables you to extend your local-area network (LAN) to other LANs at remote sites. There are more than one ways to build wide-area networks employing various types of connections, technologies, and devices.

Cisco IOS Software supports a number of WAN protocols. In this chapter, we will introduce you to High-Level Data Link Control (HDLC), Point-to-Point Protocol (PPP), and Frame Relay on serial interfaces. We will also learn how to configure these WAN protocols on Cisco routers. We also give you a brief introductions to virtual private networks (VPNs) as an alternate to traditional WAN solutions.

The OSI Layer 1 (physical layer) and Layer 2 (data link layer) work together to deliver data across a wide variety of network types. Local-Area Network (LAN) standards and protocols define how to network devices that are relatively close together, hence the term *local-area* in the acronym LAN. Wide-Area Network (WAN) standards and protocols define how to network devices that are relatively far apart, hence the

term *wide-area* in the acronym WAN. LANs and WANs both implement the same OSI Layer 1 and Layer 2 functions but with different mechanisms.

The big distinction between LANs and WANs relates to how far apart the devices can be and still be capable of sending and receiving data. LANs tend to reside in a single building or at most among nearby buildings in a campus using optical cabling approved for Ethernet. WAN connections typically run much longer distances than Ethernet LANs: across town, between cities, or even between continents. Usually only one or a few companies even have the rights to run cables under the ground between the sites. For example, a company may have two offices just across a road such that the distance between the two buildings is well within the maximum distance supported by Ethernet technologies. However, the two companies still cannot simply run a cable under the ground between the two offices due to right-of-way restrictions. When Ethernet LANs are used to connect buildings, it normally is inside a campus like a university or office complex.

Besides LANs and WANs, the term Metropolitan-Area Network (MAN) is sometimes used for networks that extend between buildings and through rights-of-way. The term MAN typically implies a network that does not reach as far as a WAN, and generally spans a single metropolitan area. However, you should keep in mind that the distinctions between LANs, MANs, and WANs are blurry. There is no set distance that means a link is a LAN, MAN, or WAN link. For example, the 1000BASE-ZX Ethernet standard with extended wavelength, single-mode (SM) fiber cabling can achieve distances upto 100 km!

A company that needs to send data over longer distances does not actually own the line or cable; it rather leases it from the company that actually own it and that's why it is called a *leased line*. The company that owns, manages, and installs such long links, or

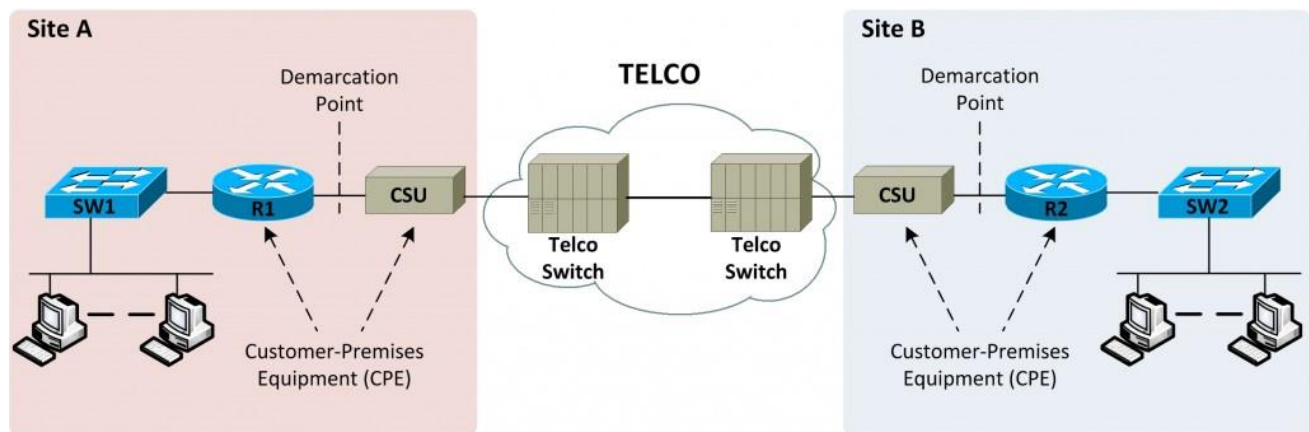
circuits has the right-of-way to run cables under streets, highways, rivers etc. The generic term *service provider* is used to refer to a company that provides leased lines for WAN connectivity.

## **11-2 Point-to-Point WANs: Layer 1**

The OSI Layer 1, or physical layer, defines the specifics of moving data from one device to another over a medium. No matter what type of data is sent, eventually the sender of data needs to actually transmit the bits to another device in the form of physical signals or waveforms. The OSI physical layer defines the standards and protocols used to make the physical transmission of bits across a network possible.

Point-to-point WAN links provide basic connectivity between two sites, as shown in Figure 12-1. In order to get a point-to-point connectivity, you would engage a service provider to install a circuit. The service provider would provision a point-to-point link or circuit and also install devices at both ends of the circuit. This kind of point-to-point WAN connection is also called a *leased circuit* or *leased line* because it is always available and you have the exclusive right to use it as long as you keep paying for it.

**Figure 12-1** Components of a Point-to-Point WAN Link



The technologies used by the service provider to build its network to support your point-to-point WAN link are complex. Fortunately, you don't need to spend time studying and learning those technologies as they are outside the scope of your CCNA exam. You can conceptually view the point-to-point WAN link as if the two routers R1 and R2 are connected back-to-back. Most of the time, all you are concerned with is the type of interface provided by the CSU that connects to your router and the speed of the leased circuit. This simplified view of a point-to-point WAN link serves you well for your CCNA exam as well as real life work as network engineer. However you will be introduced to a few basic concepts and terms related to service providers in the coming paragraphs.

Typically, the router connects to a device called a channel service unit/data service unit (CSU/DSU). The CSU/DSU is usually a standalone unit installed and maintained by the service provider and looks somewhat like an external dial-up modem. The CSU/DSU is usually placed in the same rack as the router and connects to the router with a relatively short cable, typically less than 50 feet long. The much longer cable that runs from the central office (CO) to the customer premises plugs into the CSU/DSU. Older leased line technologies used four wires or two pairs of wires but modern technologies allow the use of a single pair just like a telephone line. Sometimes the CSU/DSU is also integrated into the router and the leased line terminates directly at an interface on the router. This cable connects the CSU/DSU to the telco switch in the nearest CO and can be several kilometers long.

The router and CSU/DSU typically are two separate physical devices. However, Cisco also manufactures WAN interface cards (WICs) with integrated CSU/DSU, eliminating

the need for a separate CSU/DSU. An example is the WIC-1DSU-T1 (and WIC-1DSU-T1-V2) which is a CSU/DSU WAN interface card by Cisco for T1 or fractional T1 service. The WIC-1DSU-T1 installed in a Cisco router provides a simple and fully integrated solution from a single vendor. The WAN connectivity is provided through the standard RJ-45 interface connector on the card. The configuration is performed via the familiar Cisco IOS CLI and there is no need to learn the command syntax of an external CSU/DSU from another vendor. It obviates the external CSU/DSU affording ease of deployment, configuration, and management.

The term customer-premises equipment (CPE) is commonly used by telcos to refer to the equipment installed at the customer site. For example, the LAN switch, router, and CSU/DSU are classified as CPE in Figure 12-1.

From a legal perspective, two different companies own the various components of the equipment and lines in Figure 12-1. For instance, the router along with the cable connecting the router to CSU/DSU is typically owned by the customer. The CSU/DSU, the wiring from CSU/DSU to the CO and the gear inside the CO are all owned by the telco. The telco uses the term *demarcation point* or *demarc* to refer to the point at which the telco's responsibility is on one side and the customer's responsibility is on the other. The demarc is not a separate device, but rather a concept of where the responsibilities of the telco and customer separate. There may be different ways to establish the demarc in different countries by different service providers. In some cases, the telco also owns and manages the router in addition to the CSU/DSU. In some cases, the CSU/DSU and router are both owned by the customer. The demarc point shifts according to the specific ownership terms of a scenario. The term CPE still refers to the equipment at the customer's location regardless of ownership.

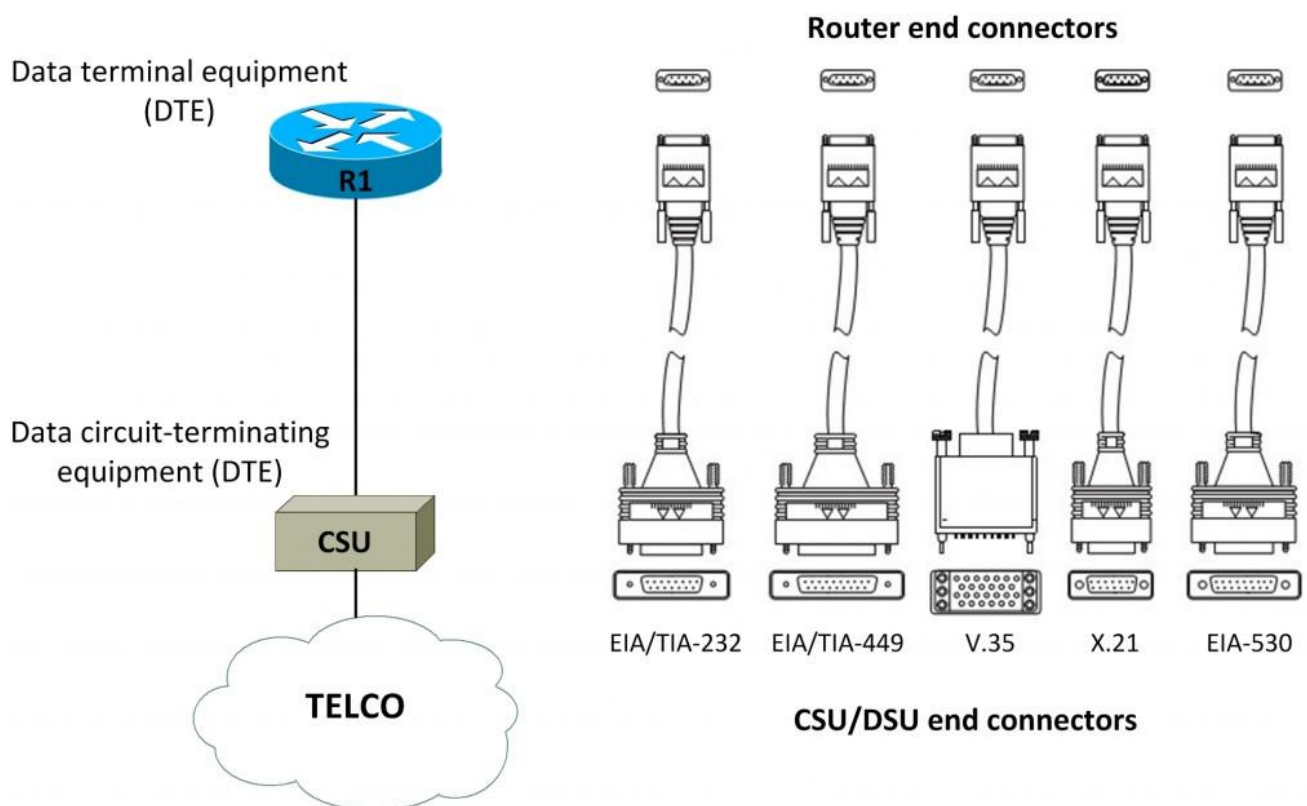
### WAN Interfaces on Cisco Routers

Cisco offers a variety of different WAN interface cards (WICs) for its routers, including synchronous and asynchronous serial interfaces. For HDLC, PPP, or Frame Relay links in this chapter, the router always uses an interface that supports synchronous serial communication.



As we discussed in the last section, leased circuits or lines are used to build point-to-point WAN links between routers. Typically synchronous serial interfaces in Cisco routers are used to connect to the CSU/DSU. The cable connecting the router to the CSU/DSU uses a connector that fits the router serial interface on the router side and a standardized WAN connector type that matches the CSU/DSU interface on the CSU/DSU end of the cable.

**Figure 12-2** Cisco Serial Connectors



As a network engineer, you have to choose the right cable based on the connectors on the router and the CSU/DSU. Beyond that you usually do not have to think about pinouts or other considerations. Once you choose the right cable and secure the connection, it just works.

## **11-3 Point-to-Point WANs: Layer 2**

We will discuss two point-to-point WAN protocols available on serial interfaces of Cisco routers namely High-Level Data Link Control (HDLC) and Point-to-Point Protocol. The two protocols are inter-related though PPP is significantly more feature-rich and advanced than HDLC.

**Table 12-1** Encapsulation Chart

Encapsulation	Leased Line	Circuit Switched	Packet Switched
HDLC	Yes	Yes	No
PPP	Yes	Yes	No
Frame Relay	No	No	Yes

We will explain how to configure leased lines between two routers, using both HDLC and PPP.

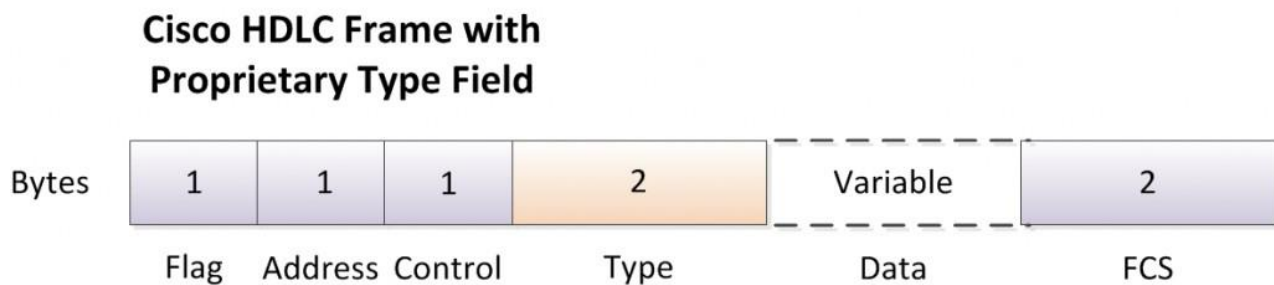
### **HDLC Concepts**

High-Level Data Link Protocol (HDLC) is a simple data link protocol that performs a few basic functions on point-to-point serial links. The standard HDLC frame does not have a protocol type field to identify the type of packet carried inside the HDLC frame. The HDLC trailer has a Frame Check Sequence (FCS) field that allows the receiving router to decide if the frame had errors in transit and discard the frame if needed.

**Key Concept** High-Level Data Link Protocol (HDLC) is the default encapsulation on serial interfaces of Cisco routers.

The absence of a protocol type field in the HDLC header posed a problem for links that carried traffic from more than one Layer 3 protocol. Cisco, therefore, added an extra **Type** field to the HDLC header, creating a Cisco-specific version of HDLC. The frame format of this Cisco version of HDLC is shown in Figure 12-4 and it is this HDLC frame found on all HDLC serial links connecting Cisco routers. Cisco routers can support multiple network layer protocols on the same HDLC link. For example an HDLC link between two Cisco routers can forward both IPv4 and IPv6 packets because the **Type** field can identify which type of packet is carried inside each HDLC frame.

**Figure 12-3** HDLC Framing



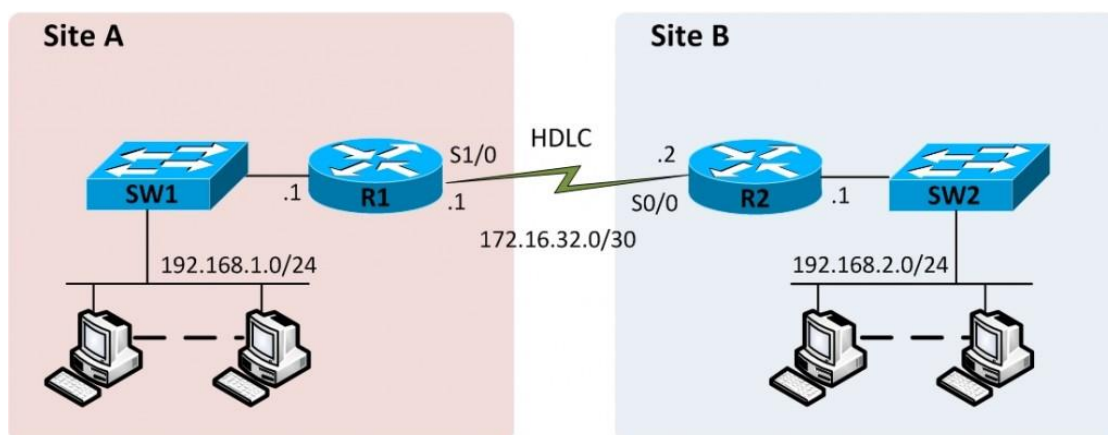
The **Address** and **Control** fields do not have much work to do these days. For example, only two routers are connected to each other on a point-to-point serial link. When a router sends a frame it is obvious that the frame is destined for the only other router on the link. You may be wondering why HDLC has an **Address** field at all. In years past, telcos offered multidrop circuits which included more than two devices with more than one possible destination, requiring an **Address** field to identify the correct destination. Both **Address** and **Control** fields had important roles in those days, but today they are not important.

## HDLC Configuration

Cisco IOS Software uses HDLC as the data link protocol, by default, on serial interfaces. In order to establish a functional point-to-point leased line connection between two routers, you first need to order a leased line. Once the leased line is provisioned, you need to complete the required cabling between routers at the two ends and CSU/DSUs. In addition to that, you just need to configure IP addresses and probably a **no shutdown** command if the interface is *administratively shutdown*. The point-to-point WAN connection would become functional with HDLC as the Layer 2 protocol.

However, many optional commands exist for serial links and we will configure a point-to-point serial links between two routers as shown in Figure 12-3, exploring some of those commands.

Figure 12-4 HDLC Configuration



First of all, let's configure the interface IP address on R1 using the **ip address** command in interface configuration mode.

If an **encapsulation** command already exists on the interface, for a non-HDLC protocol, we will have to enable HDLC using the **encapsulation hdlc** command in interface configuration mode. Alternatively, you can make the interface revert back to its default encapsulation by using either **no encapsulation** or **default encapsulation** command to disable the currently enabled protocol.

If the line status of the interface is administratively down, you must enable the interface using the **no shutdown** command. That sort of concludes our configuration. However, there are some optional commands that do not have any impact on whether our HDLC link works or not. It is always a good practice to configure a description of the purpose of the interface using the **description** command in interface configuration mode. You can also configure the speed of the link using the **bandwidth** command that takes its parameters in kbps. The bandwidth command does not set the actual bandwidth of the link which is rather determined by clocking provided by the CSU/DSU. However it is good practice to set the bandwidth equal to the actual speed of the link. Let's now go ahead and actually configure R1.

Let's now verify if the link is operational.



```
R1# show interface Serial1/0
Serial1/0 is up, line protocol is up
Hardware is M4T
Description: Serial link to R2 with HDLC encapsulation
Internet address is 172.16.32.1/30
MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
reliability 255/255, txload 3/255, rxload 3/255
Encapsulation HDLC, crc 16, loopback not set
Keepalive set (10 sec)
Restart-Delay is 0 secs
Last input 00:00:04, output 00:00:04, output hang never
Last clearing of "show interface" counters never
<Some output omitted for brevity>
```

In addition, you can also use **show ip interface brief** and **show interface description** commands to verify interface status and configuration.

The router will use a serial interface only when it is in the up/up state, as shown in the highlighted line of the output of **show** commands above. The first **up** refers to the Layer 1 status while the second **up** indicates Layer 2 status.

In a production environment, the two routers would be connected to CSU/DSUs to form a point-to-point serial link. However, in a lab environment you may need to connect two routers back-to-back without a real leased line or CSU/DSUs. There are two ways to get this done. One method is to use a device known as *modem eliminator*, which as the name says eliminated the need to have CSU/DSUs and appears to be a modem for both routers. Another method is to connect the two routers directly using back-to-back cables. In this case you have to configure clocking on one of the two routers using the **clock rate** command in interface configuration mode. This command should be used only on the one router with the DCE cable connected to it. If you are not sure which router has the DCE cable connected to it, you may use the **show controllers** command to find out.

## **11-4 PPP Concepts**

Point-to-Point Protocol (PPP) is also data link protocol used on serial links just like HDLC. However PPP is has more advanced features when compared with HDLC which is quite primitive. We will introduce you to the most important PPP concepts and proceed to do some real PPP configuration as well.

The standard version of HDLC does not have a protocol field to identify the Layer 3 protocol encapsulated by HDLC. But the PPP standard does define a protocol field to identify the type of packet inside the frame. This field allows packets from many different Layer 3 protocols to pass over a single link. However, in practice only packets for the two versions of IP (IPv4 and IPv6) are encountered. Figure 12-5 shows a PPP frame which is identical to the HDLC frame presented in Figure 12-5 earlier in the chapter.

**Figure 12-5 PPP Framing**



PPP is a versatile protocol and supports both synchronous and asynchronous links. The protocol Type field in the header allows multiple Layer 3 protocols to be carried over the same PPP link. PPP also supports authentication and two mechanisms are available for this purpose: Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP). PPP has control protocols for each higher-layer protocol supported by PPP, allowing easier integration and support for those protocols.

### PPP Components

Even though PPP framing is pretty similar to HDLC, PPP defines a set of Layer 2 control protocols that perform various link control functions. These control protocols of PPP are separated into two categories:

- **Link Control Protocol (LCP):** It has several functions related to the data link itself ignoring the Layer 3 protocol encapsulated by PPP.
- **Network Control Protocol (NCP):** There is one protocol of this category for each network layer protocol. Each protocol performs functions specific to its related Layer 3 protocol.

**Key Concept**      PPP has two components: LCP that is responsible establishing, configuring, maintaining, and terminating the connection, and NCP which is specific to the Layer 3 protocol encapsulated by PPP.

The Link Control Protocol (LCP) implements all those control functions that work regardless of the Layer 3 protocol encapsulated by PPP. All functions specific to a Layer 3 protocol are performed by the Network Control Protocol (NCP) specific to the related protocol, such as IP Control Protocol (IPCP) for the Internet Protocol (IP). PPP uses a single instance of LCP for a PPP link while one NCP instance is used for each Layer 3 protocol defined on the link. For example, a PPP link that uses IPv4, IPv6, and Cisco Discovery Protocol (CDP) will use one instance of LCP plus IPCP for IPv4, IPv6CP for IPv6, and CDPCP for CDP.

**Table 12-2** Functions of Link Control Protocol (LCP)

LCP Feature	Function	Description
Magic number	Detection of looped link	Disables the router interface if a looped link is detected so that rerouting takes place over a working route.
Link-quality monitoring (LQM)	Error detection	Disables a router interface that exceeds certain error percentage threshold, and allows rerouting over better routes.
PAP and CHAP	Authentication	Exchanges names and passwords so that each device can verify the identity of the device at the other end of the link.
Multilink PPP	Bundling multiple links	Multiple parallel PPP links are bundled together to expand available bandwidth by load balancing traffic over those links.

## Authentication

In the field of networking, authentication is a mechanism used to verify the identity of another device. This identity verification is needed to confirm that the other device is legitimate and not some one only appearing to be an authentic device in order to cause damage or steal information. For example, if R1 and R2 are to form a serial link using PPP, R1 may want R2 to somehow prove that it really is R2. This scenario is where R1 is authenticating R2, or in other words, asking R2 to prove its identity.

PPP is used over both synchronous leased lines and asynchronous dial lines, and configuration of authentication remains the same for both cases. PPP defines two authentication protocols: Password Authentication Protocol (PAP) and Channel Handshake Authentication Protocol (CHAP). Both protocols involve exchanges of messages between the two PPP speaking devices, but there are differences in detail. With PAP, the device to be authenticated starts the message exchange by sending a clear text password, claiming to be legitimate. The device at the other end of PPP link compares the password with its own password and if the password is correct, sends back an acknowledgement. The authentication process is one way and one or both devices can authenticate each other separately. PAP is simple in operation as well as configuration but it is insecure because the password is sent in clear text and can be sniffed.

Channel Handshake Authentication Protocol (CHAP) is a much more secure option than PAP and the password is never sent in clear text with CHAP. CHAP verifies the identity of the PPP peer by means of a three-way handshake. The general steps performed are:

1. After Link Control Protocol (LCP) phase is complete and CHAP is negotiated between the two devices, the authenticator sends a challenge message to the PPP peer.
2. The peer responds with a calculated through a one-way hash function of Message Digest 5 (MD5).
3. The authenticator calculates its own hash value and compares the received response against it. If the values match, the authentication is considered successful. Otherwise connection is terminated.

CHAP is a one-way authentication method, which means it involves an authenticator authenticating its peer. In practice, both peers are configured to authenticate each other and two separate three-way handshakes take place.

PAP is much less secure because PAP sends both the hostname and password in clear text inside a message. These values can be easily read if someone places a tracing tool in the circuit to sniff data. CHAP uses a one-way hash algorithm, known as MD5, with

input to the algorithm being a password that is used locally to compute the hash and never crosses the link and a shared random number.

### **PPP Phases: LCP, Authentication, and NCP**

LCP negotiation is a PPP phase in which parameters are negotiated for establishing, configuring, and testing the data-link connection. During LCP negotiation, the two routers agree whether to use PAP or CHAP for authentication or whether to use authentication at all or not. The LCP negotiation also uses a parameter called *MagicNumber*, which is used to determine if the link is looped back. A random text string is sent across the link and, if the same value is received back, the router knows that the link is looped. An LCP state of *open* means that LCP was successfully completed, while an LCP state of *closed* indicates an LCP failure.

The authentication phase is optional as PPP authentication is not mandatory. The authentication protocol agreed upon in the LCP negotiation (PAP or CHAP) is used to perform authentication in this phase.

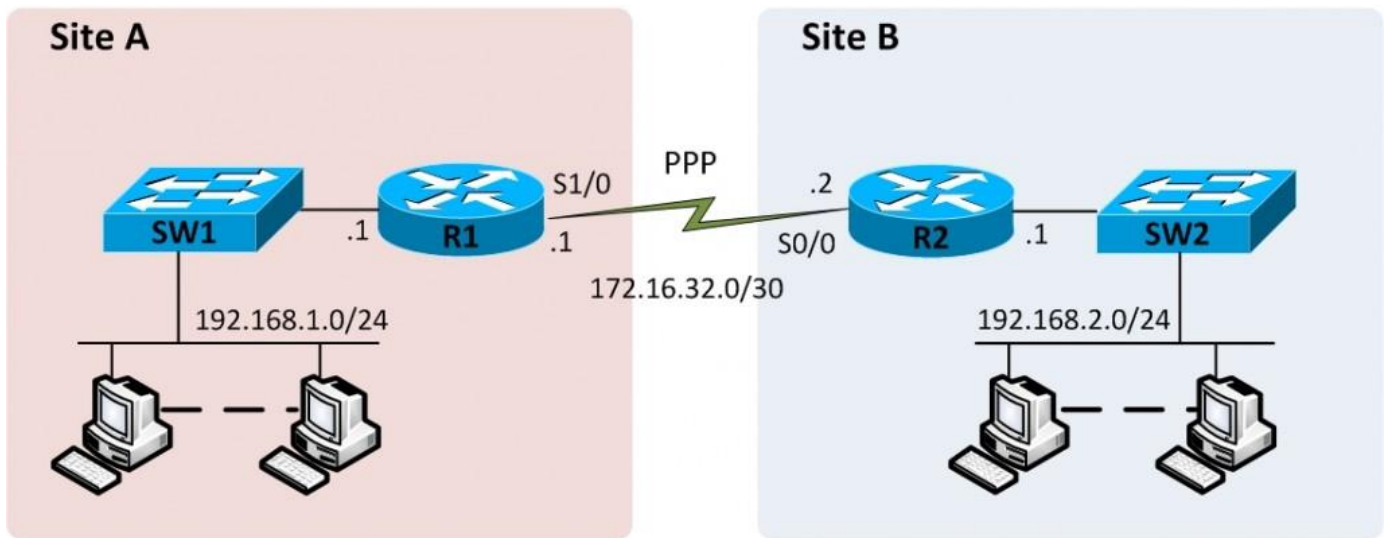
The mandatory NCP phase is used to establish and configure different network-layer protocols. The most common network layer protocol is the Internet Protocol (IP). You know that there is a specific NCP for each network layer protocol supported and the one for IP is IP Control Protocol (IPCP). The two routers exchange IPCP messages to negotiate options specific to the network layer protocol, that is, IP. IPCP negotiation can be used for IP address assignment to the peer.

## **11-5 PPP Configuration**

Point-to-Point Protocol configuration is rather straightforward if you do not configure authentication. Keep in mind that PPP authentication is optional and a link can pretty much establish without authentication. In fact, the only change here as compared with HDLC configuration earlier is that you have to use the **encapsulation ppp** command in interface configuration mode. Several other link parameters can also be configured like **bandwidth** and **description** of the interface. You may consider enabling the interface as well using the **no shutdown** command.

We will perform simple PPP configuration using two routers shown in Figure 12-5, the same internetwork used for HDLC configuration.

**Figure 12-6** PPP Configuration



Let's now configure R1 and R2 to establish a point-to-point serial link using PPP as the Layer 2 protocol.

```
R1>enable
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface Serial1/0
R1(config-if)#ip address 172.16.32.1 255.255.255.252
R1(config-if)#encapsulation ppp
R1(config-if)#no shutdown
R1(config-if)#end
R1#
```

```
R2>
R2>enable
R2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#interface Serial0/0
R2(config-if)#ip address 172.16.32.2 255.255.255.252
R2(config-if)#encapsulation ppp
R2(config-if)#no shutdown
R2(config-if)#end
R2#
```

All what we have done is to configure PPP as the encapsulation method using **encapsulation ppp** command other than configuring an IP address. That's all we need to successfully establish

a PPP serial link without authentication though. The lack of any authentication related configuration does not actually prevent the link from becoming fully operational.

Let's use the **show interfaces** command on R1 to verify if a PPP link has established.

```
R1#show interfaces Serial1/0
Serial1/0 is up, line protocol is up
Hardware is M4T
Internet address is 172.16.32.1/30
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, LCP Open
Open: IPCP, CDPCP, crc 16, loopback not set
Keepalive set (10 sec)
Restart-Delay is 0 secs
<Some output omitted for brevity>
```

The first highlighted line in the output above indicates that PPP encapsulation is being used on the interface as indicated by *Encapsulation PPP* on the first line. Also the words *LCP Open* indicate that LCP has completed its work successfully. The second highlighted line lists the fact that two NCPs, IPCP for IP and CDPCP for CDP, have also successfully been enabled. These are all positive indications that PPP is working correctly.

But that's not all about PPP configuration. We will also explore configuration for PPP authentication using one of the two available options namely CHAP. CHAP requires a password to be configured on each of the two routers R1 and R2. In fact, the password could be configured on an external AAA (Authentication, Authorization, and Accounting) server outside the router, but we will use locally configured passwords for the sake of this example.

You must configure hostnames of routers if they are not already configured using the **hostname** command in global configuration mode. In addition, you have to configure the username and password using the **username name password password** or **username name secret secret** command. The username and password are both case-sensitive. CHAP is enabled on an interface using the **ppp authentication chap** command in interface configuration mode.

```
R1#
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#username R2 password chap
R1(config)#interface Serial1/0
R1(config-if)#ppp authentication chap
```

```
R1(config-if)#end
R1#
```

```
R2#
R2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2#username R1 password chap
R2(config)#interface Serial0/0
R2(config-if)#ppp authentication chap
R2(config-if)#end
R2#
```

You may use the **show interfaces** command on either R1 or R2 to verify if PPP authentication was successful. You may also use the show users command on R1 to verify PPP authentication status as show here.

```
R1#show users
```

Line	User	Host(s)	Idle	Location
* 0	con 0	idle	00:00:00	

Interface	User	Mode	Idle	Peer Address
Se1/0	R2	Sync PPP	00:00:05	172.16.32.2

## ***11-6 Troubleshooting Serial Links***

In a perfect world, you configure a point-to-point link for HDLC or PPP and it just works. However, you may quite often find yourself in a situation when the link fails to come up while you strongly believe you configured everything right. We will briefly discuss in this section, how to isolate and fix problems on point-to-point WAN links.

A simple **ping** command is a good way to determine if a serial link configured with HDLC or PPP can or cannot forward IP packets. If you are able to successfully ping the IP address on the serial interface of the router at the other end of the link, it is enough proof that the link works.



If the **ping** does not work, you have a reason to worry. The problem may be related to functions at Layers 1, 2, or 3 of the OSI reference model. The best way to isolate the problem to one of the OSI layer is to use the **show ip interface brief** command and examine the line and protocol status.

**Table 12-3** Interface Status and Problematic Layer

Line Status	Protocol Status	Problematic Layer
Administratively down	Down	Interface is <b>shutdown</b>
Down	Down	Layer 1
Up	Down	Layer 2
Up	Up	Layer 3

Once you have identified the problematic layer, you know where to look. We introduce a few common problem you may face on point-to-point serial links.

### Keepalive Failure

The *keepalive* feature requires routers to send keepalive messages to each other, every 10 seconds by default. Keepalive messages are treated as ordinary packets, and they exist for both HDLC and PPP. The HDLC keepalive message is Cisco proprietary, whereas PPP defines a keepalive message as part of Link Control Protocol (LCP).

The *keepalive* feature enables a router notice a dysfunctional link. A router expects to receive regular keepalives from its neighboring router over an HDLC or PPP link. If a router does not receive any keepalive messages from the other routers for 5 keepalive intervals by default, the router brings down the interface, believing the router on the other end of the link is no longer working. This allows routing protocol to converge and use other valid routes if they exist.

You can change the keepalive interval from the default of 10 seconds using the **keepalive** command in interface configuration mode. It is possible to speed up failed link detection by reducing the keepalive interval. But this strategy is not useful in all situations. For example, a typical failure of serial link involves losing the Carrier Detect (CD) signal. This sort of failure is detected very quickly, within a few milliseconds. Reducing the keepalive interval cannot speed things up in this case. In most cases, the default keepalive interval is used.

You can disable keepalives using the **no keepalive** command in interface configuration mode. However, either both routers should use keepalives, or both should disable them. If there is a mistake in which one end leaves keepalives enabled while the other end disables keepalives, the link is bound to fail. This mistake only breaks HDLC links; the PPP keepalive feature can prevent the problem.

## **11-7 Frame Relay**

Frame Relay was a very popular WAN technology in the past. It still is today to some extent. However, it is safe to say that it is *being* replaced by competing technologies like Ethernet WAN, Multi-Protocol Label Switching (MPLS), and Virtual Private Network (VPN).

VPN technology has matured to a level where it is believed to provide the same level of security and confidentiality afforded by private WANs, using the Internet as transport medium. It is much cheaper to deploy VPNs over the Internet than private WANs.

The service model of MPLS is the same as that of Frame Relay. However, MPLS enables service providers to offer richer services and affords many technical advantages. MPLS is the technology of choice over Frame Relay for private WAN deployments today. Frame Relay is far from dead though due to the large existing installed base and its simplicity for point-to-point WANs connecting the branch office to corporate headquarters. Frame Relay is also used in combination with MPLS to provide Layer 2 circuits to the nearest MPLS point of presence (POP). Therefore, despite all what you have heard about Frame Relay being obsolete, it will continue to be an important networking topic for some time at least.

### **Packet Switching versus Circuit Switching**

WAN technologies can usually be categorized as either circuit-switching or packet-switching. A electrical circuit is a system of conductors (wires) forming a complete path around which a current can flow. The original telephone systems actually created an electrical circuit between two phones in order to carry the voice signal. The leased lines used for carrying data are also circuits, providing the ability to transfer bits as signals between two end points. In telecommunications terminology today, a circuit refers to the physical path between two end points providing the ability to send voice or data from one end point to the other.

Packet switching, as a technology, is more complex than circuit switching. The devices involved in packet switching have to do more than simply passing bits as signals from one end point to another. The devices in the service provider's network have to be intelligent for packet switching. This is in contrast to circuit switching where devices in the service provider's network simply have to carry signals without making sense of them. With packet switching, the devices read the bits sent by customers interpreting usually some form of address field in the packet header. The address field in the packet header is used by the devices to make choices, switching one packet to go in one direction and the next packet to go possibly in another direction to another device.

Circuit switching is an old but expensive technology, and it is what the traditional telephone network known as the public switched telephone network (PSTN) uses. Packet switching is more modern and may eventually replace circuit switching completely. Meanwhile we have to live in a world that is a hybrid of the two technologies.

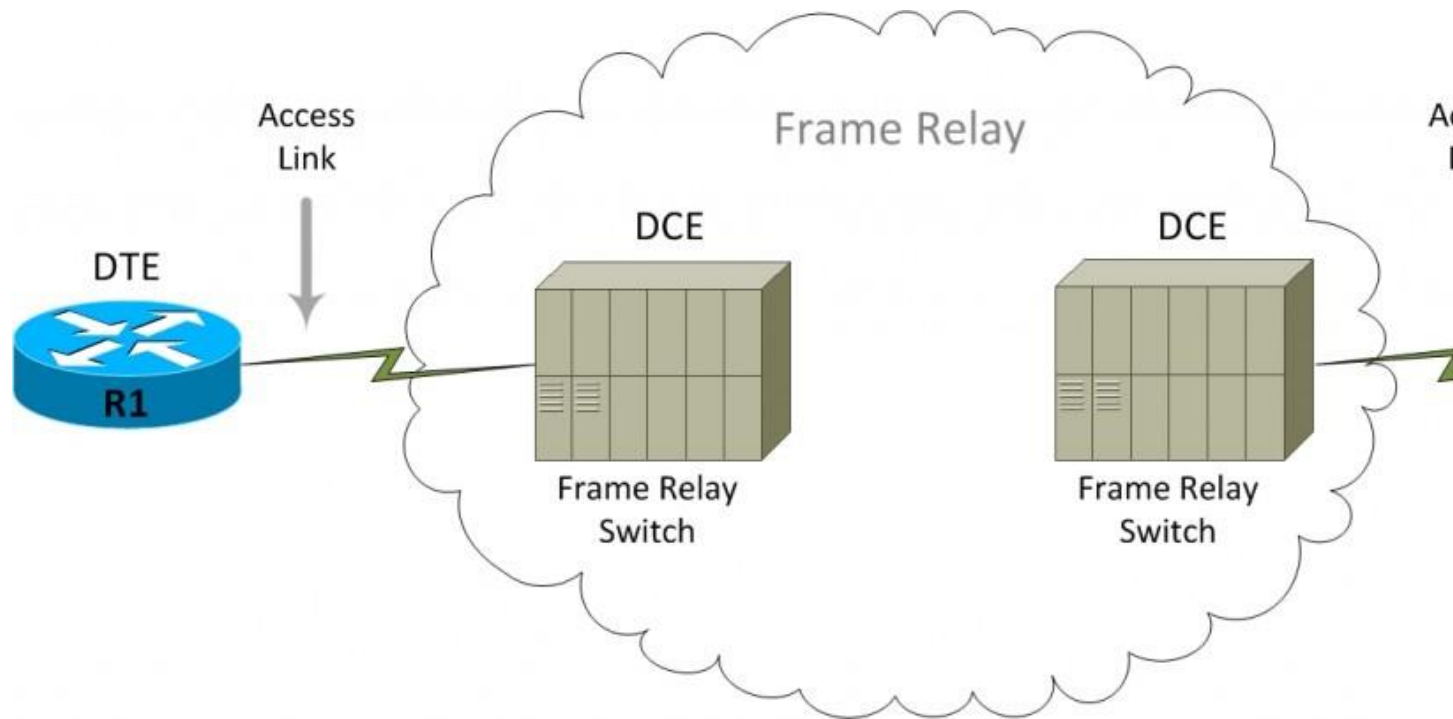
We will now cover Frame Relay thoroughly describing terminology, protocol details, and configuration.

### **Frame Relay Concepts**

Frame Relay is more complex a technology than point-to-point WAN links but also provides more features and benefits. Frame Relay networks are multiaccess networks, which means that more than two devices can connect to the network. This is similar to LANs where more than two devices can attach to the same network and any two devices can communicate directly. However, unlike LANs you cannot send a broadcast at data link layer over Frame Relay. Therefore, Frame Relay networks are termed as non-broadcast multiaccess (NBMA) networks.

Figure 12-7 presents a Frame Relay topology showing its most basic components.

**Figure 12-7** Frame Relay Components



A Frame Relay network is made up of a large number of Frame Relay switches dispersed all over the coverage area of a Frame Relay service provider. This coverage area may span a country, region, or even the whole world. The switches are interconnected in a complex mesh topology. Some Frame Relay switches also terminate user circuits, in addition to connecting to other switches, and are called access switches. Other Frame Relay switches do not terminate user circuits, connecting to other Frame Relay switches only, and make the backbone of the Frame Relay network.

A leased line is installed between the router at a customer site and the nearest Frame Relay switch. This leased line is called the *access link*. In the context of Frame Relay, the router is the data terminal equipment (DTE) while the Frame Relay switch is the data circuit-terminating equipment (DCE). To ensure that the link is working DTE and DCE exchange regular messages with each other. These keepalive messages, along with other messages, are defined by the Frame Relay Local Management Interface (LMI) protocol. Please keep in mind that the terms DTE and DCE have different meanings in different contexts and the terms here are used in the context of Frame Relay.

The physical connectivity from a Frame Relay DTE router to the Frame Relay network is the access link. However, the end goal is to provide end-to-end connectivity between two DTE routers across the Frame Relay cloud. The logical end-to-end communications path between two DTE device is known as a virtual circuit (VC). The provisioning of virtual circuits is

responsibility of the service provider, and these predefined virtual circuits are also known as permanent virtual circuits (PVC). Frame Relay routers use the data link connection identifier (DLCI) as the Frame Relay address. DLCI identifies the VC over which the frame should travel.



**Key Concept** Committed Information Rate (CIR) is the average rate, in bits per second, at which Frame Relay switch agrees to transfer data for a customer.

Let's now formally define some important Frame Relay terms before moving forward:

- **Virtual circuit (VC)** is a logical communications path that is used by frames travelling between DTEs.
- **Permanent virtual circuit (PVC)** is a permanently defined virtual circuit. PVC is analogous to a point-to-point leased line in concept.
- **Switched virtual circuit (SVC)** is set up dynamically when needed. An SVC is analogous to a dial-up connection in concept.
- **Data terminal equipment (DTE)** is a networking device like a router used by a customer to connect to the Frame Relay network of a service provider. The DTE typically resides at the customer site and is frequently referred to as customer premises equipment (CPE).
- **Data circuit-terminating equipment (DCE)** are the Frame Relay access switches that terminate customer access links and reside in the service provider network. The term DCE is also considered to mean data communication equipment by many.
- **Access link** is the leased line between the DTE (router) and DCE (Frame Relay switch).
- **Access rate (AR)** is the speed at which the access link is clocked. The access rate does not necessarily have to match the CIR. However in order to fully utilize the CIR, the access rate must be equal to or higher than the CIR.
- **Committed information rate (CIR)** is the speed at which the bits can be sent over a VC, according to the service contract between the Frame Relay service provider and its customer.
- **Data link connection identifier (DLCI)** is a Frame Relay address present in the header of every Frame Relay frame. DLCI is significant over a single hop only and different DLCI values may be used on different hops along a VC for the same packet.
- **Non-broadcast multi-access (NBMA)** is a network on which broadcasts are not supported but more than two device can be connected to the same network.
- **Local Management Interface (LMI)** is the protocol used between a DCE and DTE to manage the connection. LMI involves messages to establish SVCs, status messages for PVCs, and keepalives to mention a few.

## Virtual Circuits

Frame Relay is a cost-effective alternate to point-to-point leased lines to build enterprise WANs. In the absence of Frame Relay, enterprises wishing to connect offices worldwide would have to lease very expensive international leased circuits to connect LANs through routers. A Frame

Relay network is owned by a service provider offering services to companies that want to connect its locations to each other. Frame Relay virtual circuits act like point-to-point leased lines for the customer while providing significant cost benefits as compared with leased lines.

**Figure 12-8** Frame Relay Virtual Circuit (VC)

A virtual circuit (VC) spans the access links at the two ends as well as the Frame Relay network. For example, you can see two VCs in figure 12-8, one between R1 and R3 and the other between R2 and R3. Bold and grayed dashed lines have been used to represent VCs. You should keep in mind that the Frame Relay network is owned and operated by a service provider and is shared by many customers of the same service provider. Yet virtual circuits provisioned by the service provider for a certain customer create the illusion of a point-to-point dedicated circuit. Also the traffic from different customer is kept separate and Frame Relay networks built around this model are considered sufficiently secure.

Originally, when the world was moving from expensive private leased lines to the co-operative model of Frame Relay, customers were concerned about bandwidth because of the contention within the Frame Relay cloud with other customers for available capacity. In order to address these concerns, Frame Relay uses a concept of committed information rate (CIR). Each VC has a CIR, which is a guarantee by the provider that a particular VC would get that much bandwidth. So you can migrate from a private leased line to Frame Relay with a CIR equal to the leased line bandwidth.

Frame Relay service model requires one access from each site to the Frame Relay service provider, regardless of the number of sites to be interconnected. This is not the case if you want to build a WAN using private leased lines. In that case you would need  $N*(N-1)$  leased lines where N is the number of sites you are trying to connect. For example, if you have 3 sites you would require  $3*(3-1)=6$  leased lines, while for 10 sites the number of leased lines required steps up to  $10*(10-1)=90$  leased lines. This solution simply does not scale to large deployments. Though the access links required to connect a site to nearest Frame Relay point of presence (POP) are still private leased lines, but they are shorter and fewer.

When a Frame Relay network is designed, there may not be a VC between any pair of sites. If there is a PVC between any two sites, it is called a full-mesh topology. When not all pairs of sites have a direct PVC, it is called a partial-mesh topology. In most practical scenarios, partial mesh is used as not all customer sites typically need to connect to all other sites. For example, global enterprises typically use a star-topology which is a special case of partial-mesh topology. In a star topology a large number of remote branch offices are connected to the data center to access the resources including data and applications.

## **11-8 LMI and Encapsulation Types**

While the PVC is a point-to-point logical path between two customer routers, there are many physical and logical components that work together to create the illusion of a single logical path. Each router needs a physical access link from the router to the nearest Frame Relay switch. The provider needs to have some kind of physical network between those switches as well. In addition, the provider has to somehow provision those virtual circuits in order to make sure frames sent from one end of a VC arrive at the correct destination.

Frame Relay uses the Local Management Interface (LMI) protocol to manage each physical access link and the PVCs that use that link. The basic Frame Relay protocol format used for carrying user data frames is also used to carry LMI messages. However, LMI messages are sent in frame distinguished by a special LMI-specific DLCI usually set to 1023.

Two LMI message types have been defined that flow between the router acting as DTE and the Frame Relay switch acting as DCE. The Status-enquiry messages are sent from the router to the switch and allow the router to ask about the status of network. The Status messages are sent from the switch to the router responding to status-enquiry messages. In fact, the Frame Relay switch sends two types of messages: a status message every 10 seconds and a full status message instead of a status message every 60 seconds. The full status message contains all the information about known DLCIs and their state. The LMI status-enquiry messages are sent every 10 seconds from the router to the switch while the router responds with the status message. These periodic LMI messages also serve as *keepalives* for both the router and the switch. LMI status messages act as a keepalive between the DTE and DCE. If the access link is having a problem, these keepalives will be missed and link problem will be detected. In addition to performing a keepalive function between the DTE and DCE, LMI status messages also signal if a PVC is active or inactive. Every PVC is predefined by the Frame Relay service provider, but its status can change due to network conditions like failure of trunk links in the provider network. An access link may be up and running and keepalives may be present but one or more VCs may still be down. The reason is that a VC is an end-to-end logical connection that involves not only the access links at the two ends but also spans the core of the provider network. The router needs to know that which VCs are functional and which are not. The router learns this information as well from the Frame Relay switch through LMI status messages.

In addition to the common features like keepalives, LMI has several optional features defined as LMI extensions. We will briefly introduce two LMI extensions related to global addressing and multicasting. The basic Frame Relay specification supports DLCI values that are only locally significant. For example, the DLCI value used on the access link between a router and Frame Relay switch is significant only on the access link and does not in any way identify the router globally. This DLCI value cannot serve as an address for the router due to its local significance. In other words, Frame Relay addresses do not exist and hence cannot be discovered by usual address resolution methods. Therefore, static maps must be created to tell a router which DLCI to use to reach a remote router. The global addressing extension solves this problem by allowing DLCI values that are globally significant and hence can serve as addresses of individual end routers. The Frame Relay network with global addressing looks much like a LAN to the end routers that can use global addresses (DLCIs) as Frame Relay addresses similar to MAC addresses used in a LAN.

The multicasting extension defines multicasting as another optional LMI feature. There is a series of four reserved DLCI values (1019 to 1022) that represent multicast groups. The frames sent by a device using one of these reserved DLCIs are replicated by the network and sent to all destinations in the group. The LMI extension for multicasting also defines LMI messages to notify devices of the presence, addition, and deletion of multicast groups.



Cisco routers have three options for different variations of LMI protocols: Cisco, ITU, and ANSI. These LMI options have their differences and are incompatible with each other. For LMI to work correctly, both the DTE and DCE devices across an access link must use the same LMI type.

LMI configuration is pretty straightforward. Most of the time, we are good with the default LMI setting. This default setting uses something known as LMI **autosense**, in which router simply figures out on its own which LMI type the Frame Relay switch is using. You can just let the router autosense the LMI and never bother manually configuring it on the router. However, if you choose to configure the LMI type manually, it will automatically disable the autosense feature. Table 12-2 lists the three LMI types, the standard document, and the keyword used in the Cisco IOS Software **frame-relay lmi-type** interface configuration mode command.

**Table 12-4** LMI Types

LMI Type	Standard Document	Cisco IOS Keyword
Cisco	Proprietary	<b>cisco</b>
ANSI	T1.617 Annex D	<b>ansi</b>
ITU	Q.933 Annex A	<b>q933a</b>

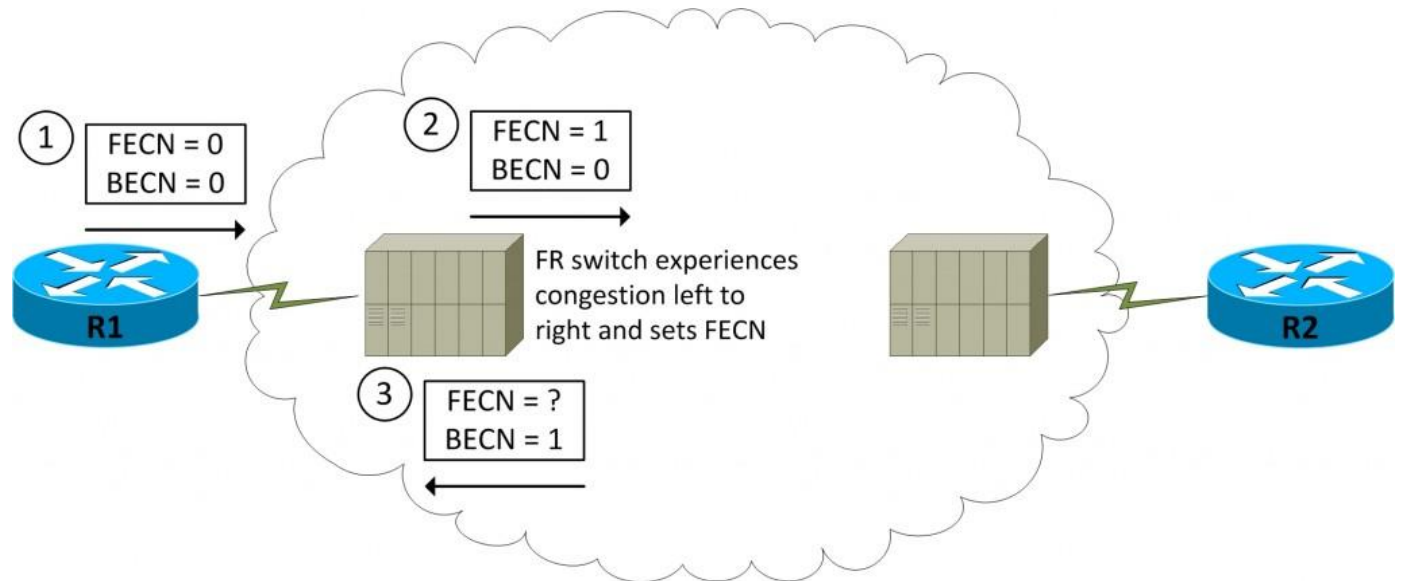
## **11-9 Frame Relay Congestion Control**

There are three flag bits inside the Frame Relay header that can be used to control what goes on inside the Frame Relay network. Imagine a situation when one (or more) Frame Relay sites/routers use an access link that is clocked higher than the CIR of a VC. In such a situation, the router can send more data to the Frame Relay switch at the edge of the provider network than what is allowed by the contracted rate or CIR between the customer and service provider. The

three bits in the Frame Relay header can influence how the switches control the network when the network gets congested due speed mismatches. These bits are:

- Forward Explicit Congestion Notification (FECN)
- Backward Explicit Congestion Notification (BECN)
- Discard Eligibility (DE)

**Figure 12-9** Operation of FECN and BECN



The FECN bit can be set by a router as well as a Frame Relay switch to indicate that the frame itself has experienced congestion. In other words, FECN indicates that congestion exists in the direction in which the frame is travelling. Keep in mind that network congestion can be unidirectional. In other words, the network can become congested in one direction while not being congested at all in the other direction. Referring to the Figure, router R1 sends a frame out to the switch with both FECN and BECN set to zero, shown as Step 1. The switch on the left experiences congestion left to right, and sets the FECN bit to 1 before sending the frame out, shown as Step 2. But what's the point of all this? The goal is to somehow make R1 reduce the speed at which it is sending frames in view of the congestion. But R1 needs to be informed of network congestion before it can think of slowing down. The Frame Relay switch on the left, knowing that it set FECN in Step 2, can now set the BECN bit in the next frame going right to left toward R1 on that same VC, shown as Step 3. When R1 receives a frame with BECN set, it knows that congestion occurred in the opposite direction. In other words, the BECN bit set in a frame received by R1 says that congestion occurred for the frame sent by R1 on the same VC (to R2). R1 can then decide to slow down a bit (it's a choice not compulsion for R1).

The IOS feature used by R1 to slow down is known as Traffic Shaping. It essentially makes R1 send some packets, wait a while, send some more packets, wait again, and so on. If the router keeps sending non-stop, it would be sending frames at the access rate or the clock rate of the

access link. By the wait periods introduced by Traffic Shaping the router effectively sends at a rate lower than the access rate. We can configure Traffic Shaping with the appropriate parameters to even make the router send exactly at the CIR when the access rate is higher than the CIR.

Finally, the Discard Eligibility (DE) bit allows the provider to selectively discard frames in which the DE bit is set at times of congestion. Frame Relay service providers usually build their networks to handle traffic loads that far exceed the collective CIRs of all VCs. As a result, customers may be allowed to send data at rates higher than the CIR. However, if one or more customers start sending data that way exceeds their contracted CIR, the provider can rightfully discard some traffic sent by those customers. The provider can set the DE bit on some frames received by such a customer that exceed the CIR. The marked frames are not discarded when there is no congestion. However when network congestion does happen, these DE marked frames are the first to be dropped. When the switch marks some of the frames received by a customer, it would normally do it indiscriminately. Some high priority frames sent by a customer may get marked and dropped ahead of some low priority frames. The customer may also want to set the DE bit in some frames, such as for less important traffic. The customer can ensure that the more important traffic gets through the Frame Relay network, even when the provider has to discard traffic. When the provider's network is not so congested, the customer can pump a lot of extra data through the network without its being discarded.

## **11-10 Frame Relay Encapsulation**

Frame Relay is a data link protocol and the customer router encapsulates each Layer 3 packet inside a Frame Relay frame comprising a header and trailer before it is sent out the access link. The header and trailer used is actually defined by the Link Access Procedure Frame Bearer Services (LAPF) specification, ITU Q.922-A. That was quite a mouthful but the LAPF framing,

shown in Figure 12-9, provides important functionality including error detection with the FCS in the trailer and a DLCI field along with a few other fields in the header.

**Figure 12-10** LAPF Framing



The standard LAPF header is too simplistic and does not provide all the fields needed by Frame Relay routers. More specifically, there is no Protocol Type field in LAPF. Each data link layer needs such a field to define the type of Layer 3 packet carried by the data link frame. If Frame Relay uses only LAPF header, routers cannot support multiprotocol traffic because there is no way to identify the type of Layer 3 protocol.

The simple LAPF header was extended to compensate for the absence of a Protocol Type field:

- Cisco created a proprietary additional header, which appears between the LAPF header and the Layer 3 packet shown in Figure 12-10. It includes a separate 2-byte Protocol Type field with values exactly matching the ones used in the same field Cisco uses for HDLC, as discussed earlier in the chapter.
- Internet Engineering Task Force (IETF) defined the second solution via RFC standards 1490 and later 2427. This solution is known as Multiprotocol Interconnect over Frame Relay and it defines a header similar to the Cisco proprietary solution placed between the LAPF header and Layer 3 packet. The additional header includes a Protocol Type field as well as several other options.

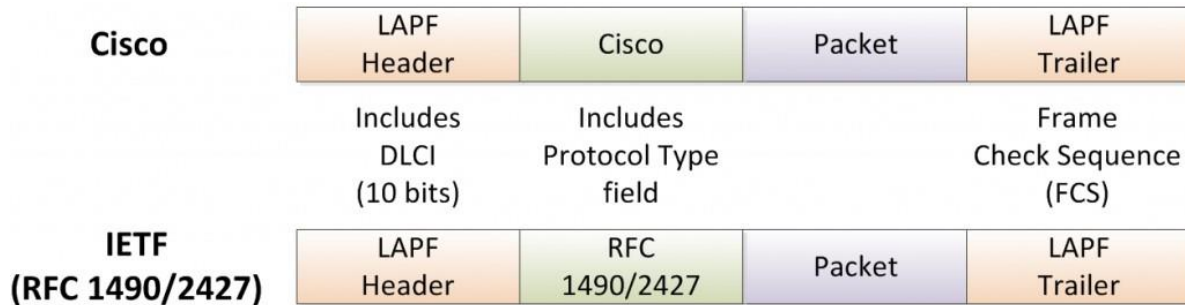


**Key Concept** Frame Relay encapsulation has two types: Cisco which is proprietary and the default on Cisco routers, and IETF which is standards based. Cisco encapsulation can be used when all routers are Cisco while IETF can be used in a multi-vendor environment.

You should keep in mind that Frame Relay encapsulation should match on the routers at the two ends of a VC. If you fail to match the Frame Relay encapsulation (both sides **cisco** or both **ietf**) on the two routers, the connection does not come up. However, if you have Cisco routers at both ends of the connection (a likely scenario), and you don't explicitly configure Frame Relay encapsulation, both routers default to **cisco** and the connection does get established. Frame Relay switches do not care about the Frame Relay encapsulation. In Cisco IOS Software

configuration, the Cisco proprietary encapsulation is called **cisco** while the other one is called **ietf**.

**Figure 12-11** Cisco and IETF Framing



## **11-11 Frame Relay Addressing**

Frame Relay defines how to deliver frames from one router to another across the Frame Relay network. The router uses a single physical access link to connect to the Frame Relay switch. The single access link may have many VCs connecting it to many remote routers. There must be something to identify each of the remote routers. That something is the data-link connection identifier or DLCI – the Frame Relay address.

The DLCI is a 10-bit value written in decimal. The possible range of DLCIs is 0-1023, however the low- and high-end values are usually reserved and typical DLCI values range from around 17 to a little less than 1000.

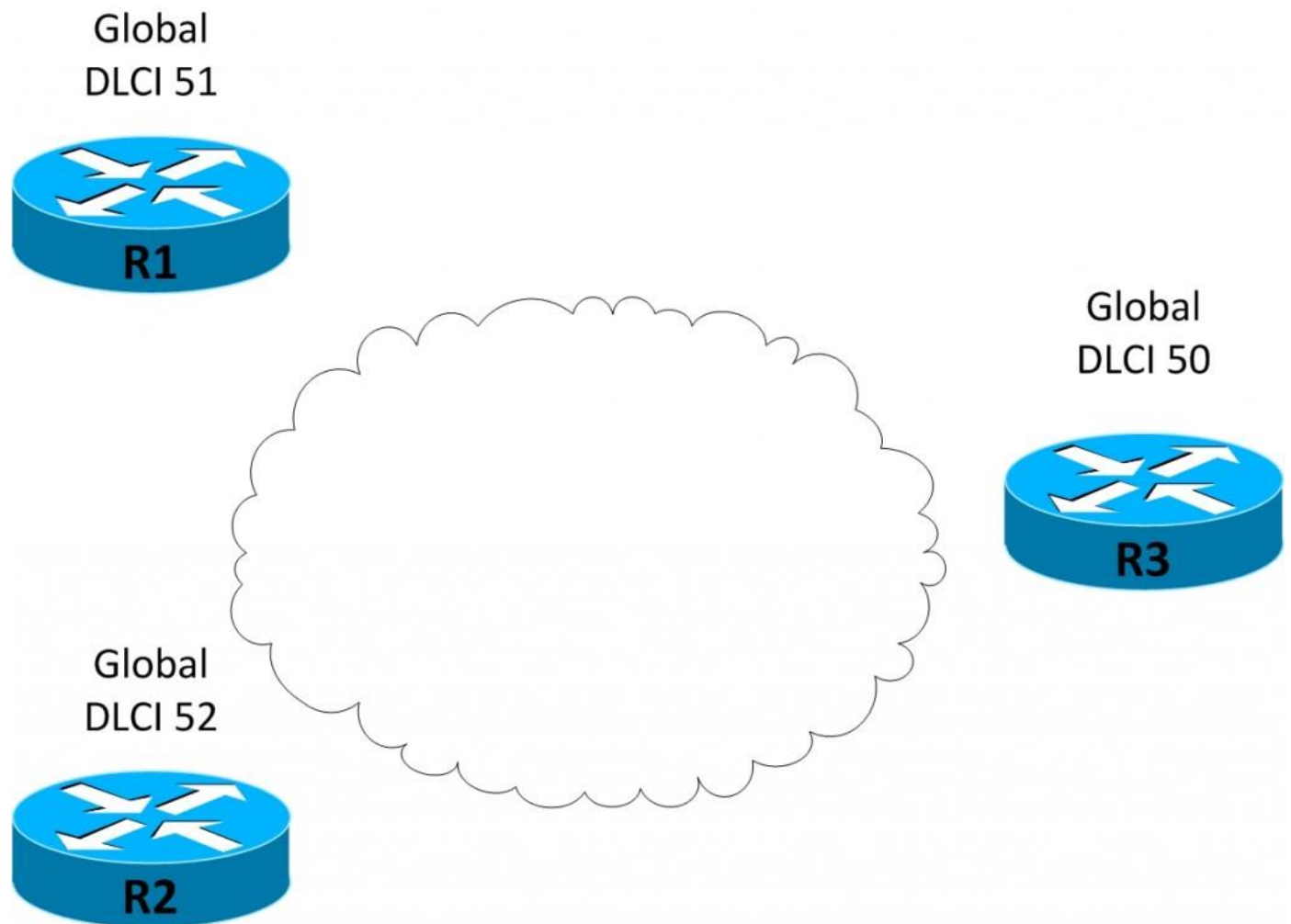
DLCIs can be simple and confusing at the same time. The most important fact about the DLCI is that it does not identify a VC but only a single hop on the VC. A Frame Relay service provider assigns two local DLCI values to each PVC: one for each end of the PVC to be used between the DTE router and DCE switch.

### **Frame Relay Global Addressing**

Global addressing is a Frame Relay addressing scheme that serves to lessen the confusion about DLCIs. Global addressing makes DLCIs look like MAC addresses in Ethernet LANs. Global addressing is a very simple convention of how DLCI values are assigned when planning a Frame Relay network so that working with DLCIs is much easier. Global addressing does not change anything inherently with DLCIs or Frame Relay addressing. It simply chooses such DLCI values that they become more intuitive to understand and deal with.

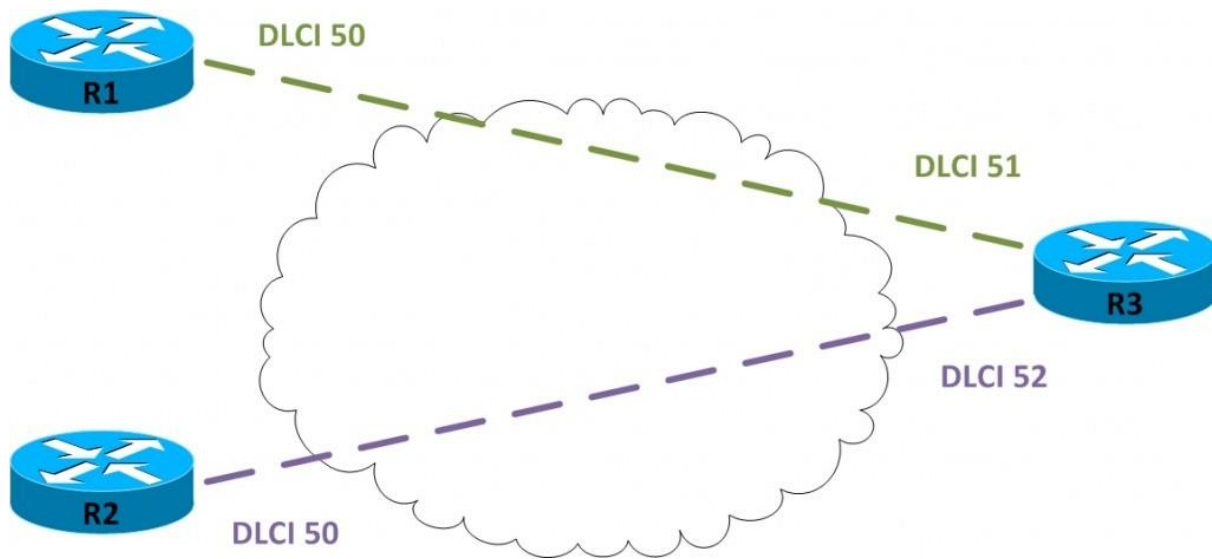
Here is how global addressing works. The Frame Relay service provider supplies a configuration sheet and a diagram similar to Figure 12-x, with global DLCIs shown.

**Figure 12-12** Frame Relay Global DLCIs



Frame Relay global addressing as planned in Figure 12-12 is used to place DLCIs in Frame Relay frames as shown in Figure 12-13. For example, router R1 uses DLCI 50 when sending a frame to router C, because router R3's global address is 50. In a similar manner, router R3 uses DLCI 51 when sending frames over the VC to router R1. The beauty of global addressing is that it works like addressing in a LAN with a single MAC address for each device, making it much more logical to most people.

**Figure 12-13** DLCI Values for Two PVCs



In Figure 12-13, the PVC between routers R1 and R3 has two DLCIs assigned by the service provider, one at each end. R1 uses local DLCI 50 to identify the PVC and R3 uses local DLCI 51 to identify the same PVC. Similarly, the PVC between routers R2 and R3 also has two DLCIs assigned, one at each end. In this case, R2 uses local DLCI 50 while R3 uses local DLCI 52.

DLCI values are only locally significant and can be reused on different links. In Figure 12-11, both R1 and R2 use DLCI 50 to identify their respective PVCs which is perfectly fine. However, the local DLCIs on a *single* access link must be unique among all PVCs that exist on that access link. If you work for an enterprise, you need not worry about DLCIs as their values are chosen by the provider.

The local router is aware of only the local DLCI and it effectively identifies a PVC for the router. When you configure a router, you only configure the local DLCI value and don't need to concern yourself with DLCI value at the other end of the PVC.

The Frame Relay header lists only a single DLCI field which performs the addressing function. It does not identify both a source and destination address like the Ethernet and IP headers. The Ethernet header has both a source and destination MAC address, while the IP header contains both source and destination IP addresses.

The DTE router identifies a PVC with the DLCI assigned to that VC by the provider. The DTE router will send all packets for that VC encapsulated in a Frame Relay frame with that specific DLCI value listed in the frame header. The service provider itself assigns DLCI values to the customer and it knows which DLCI values are to be used at the two ends of a VC to enable end-to-end communication on a VC.

Frame Relay networks have some additional considerations when it comes to assigning subnets and IP addresses on interfaces. You can have:



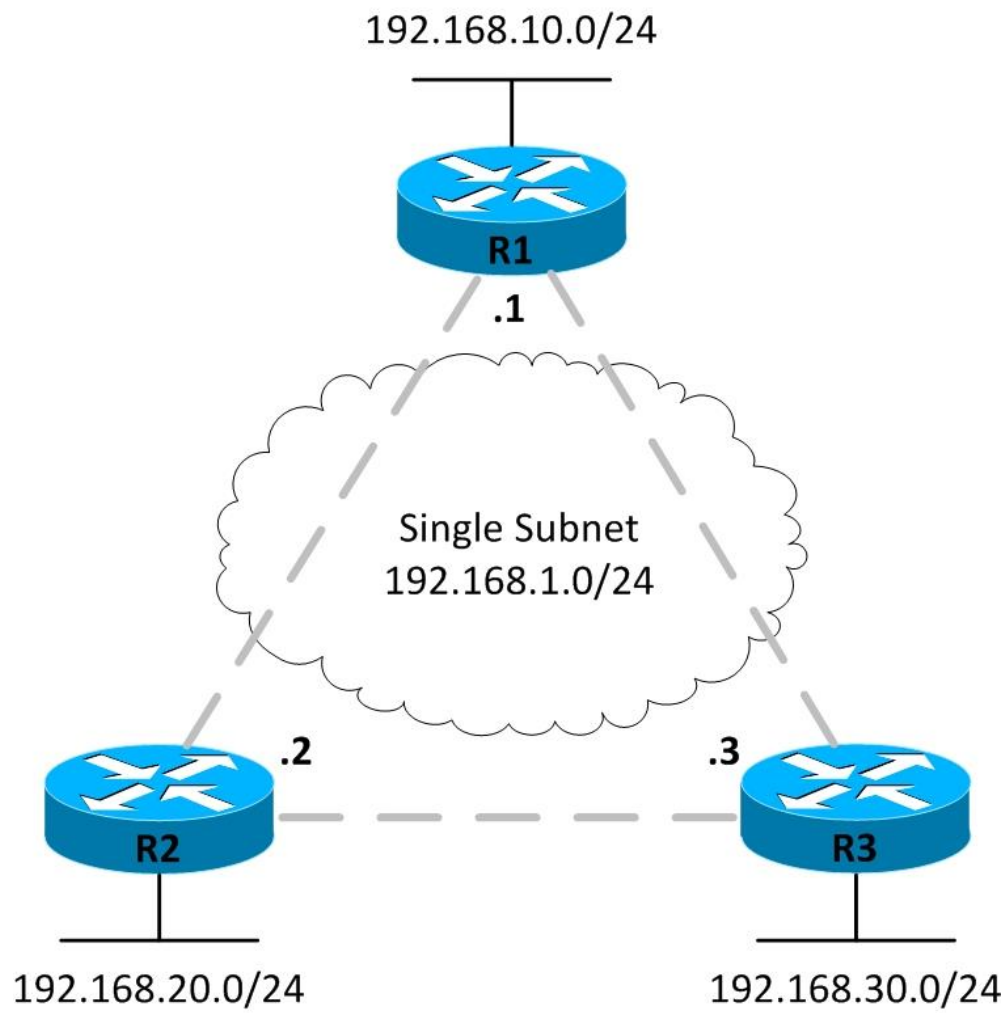
- Single subnet covering all Frame Relay DTEs
- One subnet per VC
- A hybrid of the first two options

## **11-12 Frame-Relay Topology Approaches**

### **Single Subnet for all Routers**

The first approach is to use a single IP subnet for the whole Frame Relay network, as shown in Figure 12-12.

**Figure 12-14** Single Subnet for all Routers

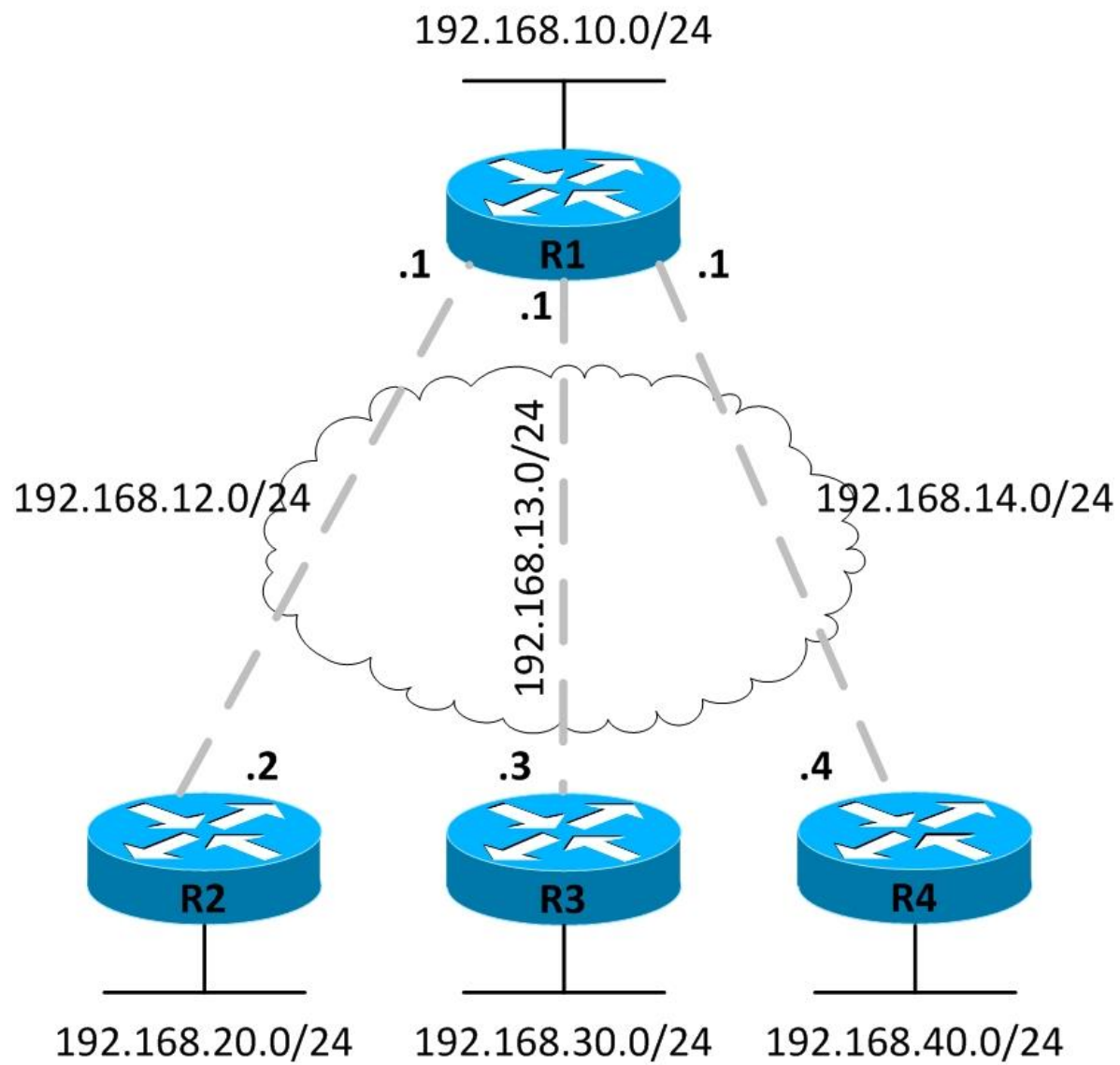


The single-subnet option is normally used when there is a full mesh of virtual circuits (VCs). In a full mesh, every router has a virtual circuit to every other router, which means that every router can send frames directly to every other router. This addressing scheme resembles Ethernet LANs with the difference that IP addresses are configured on the serial interfaces of routers with Frame Relay encapsulation. The single-subnet option is conceptually simple because it looks like what you are used to on Ethernet LANs. However, the vast majority of Frame Relay deployments use partial mesh and the single-subnet option is not well suited for that.

### **One Subnet per VC**

The second alternative of having one IP subnet per VC, works better for a partially meshed Frame Relay network, like the one shown in Figure 12-13. This is the more prevalent Frame Relay network because most organizations have a large number of remote sites that need to connect to a central site to access applications. Here there is no VC, for example, between R2 and R3 and so R2 cannot communicate directly with R3.

**Figure 12-15** One Subnet per VC



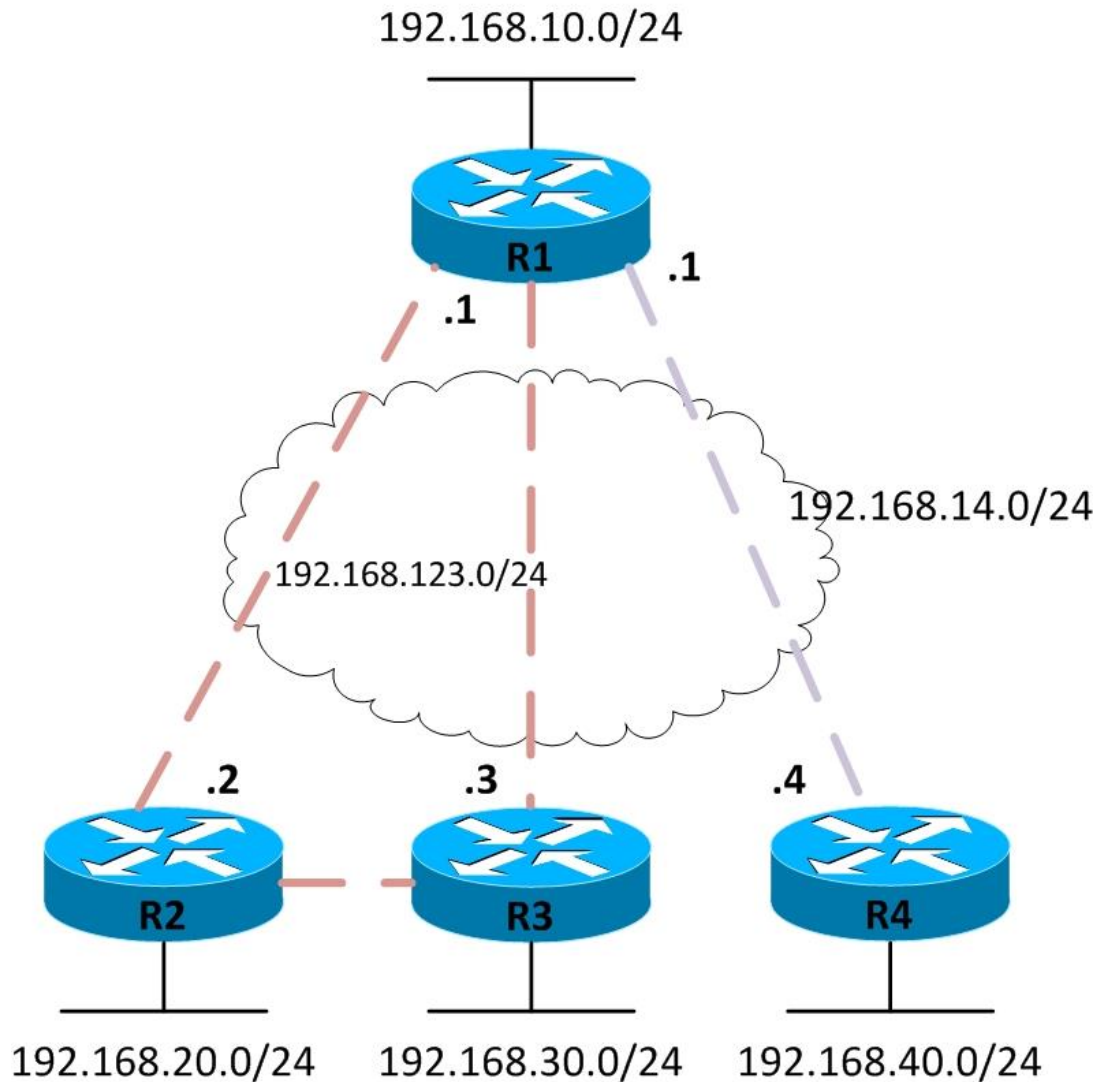
You may have noticed that R1 has three IP addresses associated with it. Cisco IOS software allows you to create logical subdivisions of a physical interface, called subinterfaces. Subinterfaces allow R1 to have three IP addresses associated with the same physical interface. The router can treat each subinterface and the VC associated with it as a separate point-to-point serial link.

Also, we are using private IP addresses with predictable /24 prefixes to enable you focus on underlying concepts rather than numbers. However, you should keep in mind that on point-to-point subinterfaces you would usually see /30 addresses with 255.255.255.252 as subnet mask. This allows for only two valid IP addresses on a subnet and conserves available IP address space.

### **A Mix of Full and Partial Mesh**

The third and last alternative for IP addressing is a mix of the first two alternatives. Figure 12-14 show a trio of routers R1, R2, and R3 with VCs in full mesh among them while a single VC to R4. In this case, you have two options for Layer 3 addressing. The first is to treat each VC as a separate Layer 3 subnet. However you would need four subnets for the Frame Relay network in that case. The second option also shown here is to create a smaller full mesh between routers R1, R2, and R3 while leaving R4 out. This allows R1, R2, and R3 to use a single subnet, The VC between R1 and R4 is then treated as a separate subnet, which results in only two subnets for the Frame Relay network rather than four.

**Figure 12-16** A Mix of Full and Partial Mesh



In order to accomplish this addressing scheme, subinterfaces are used. Point-to-point subinterfaces are used when a single subnet is mapped to a single VC, for example, between R1 and R4. Multipoint subinterfaces are used when more than two routers are in the same subnet, for example, with R1, R2, and R3.

Multipoint interfaces can terminate more than one VC, and the term multipoint refers to the fact that more than one remote sites may be reachable off the interface.

We will provide you full configurations for all three scenarios discussed so far in the next section.

## **11-13 Frame Relay Configuration**

You should have a good understanding of Frame Relay by now and its time to get your hands dirty with some configuration. Frame Relay configuration has any options, yet the actual configuration you perform can be very basic depending on how many default settings can be used. Cisco IOS Software uses the following defaults for Frame Relay:

- **LMI** Cisco IOS automatically senses the LMI type by default and this feature is referred to as LMI autosense. If you manually configure the LMI using the **frame-relay lmi-type** command, LMI autosense is silently disabled.
- **IARP** Cisco IOS automatically discovers the next-hop IP address associated with a DLCI or VC using Inverse Address Resolution Protocol (IARP). You can also create a mapping between a DLCI and next-hop IP address manually using **frame-relay map ip** command.
- **Encapsulation** Cisco IOS uses Cisco encapsulation for Frame Relay and if you are using only Cisco routers, this default setting works fine without any additional configuration.



You are familiar with the concept of physical and logical sub-interfaces. For example, you may configure several sub-interfaces on a single Fast Ethernet physical interface on a Cisco router. Frame Relay is a Layer 2 WAN protocol that can be configured on physical serial links. In addition to physical interfaces, you can also configure two types of logical interfaces for Frame Relay – **point-to-point** and **multipoint**. We will introduce you to some of the specifics of Frame Relay configuration for these different interface types.

In certain cases, you may have a working Frame Relay connection by just using a single command **encapsulation frame-relay**, and leaving everything else to default values. However, you should be familiar with the many configuration options and when they are used. Frame Relay is the source of many tricky questions on CCNA, CCNP, and beyond.

Here is your step-by-step guide to configuring Frame Relay:

- The first step should always be to configure the physical interface to use Frame Relay encapsulation using the command **encapsulation frame-relay** in interface configuration mode.
- Configure an IP address on the interfaces or sub-interface using the good old **ip address** command.
- Optionally, configure the LMI type of each physical interface using the **frame-relay lmi-type** command.
- Optionally, change the default Frame Relay encapsulation using the command **encapsulation frame-relay**. If you use the command on the interface (or sub-interface), it will change the encapsulation for all VCs on the interface (or sub-interface). If you want to change the encapsulation only for a specific VC, you should use the **ietf** keyword with the command **frame-relay interface-dlci** (point-to-point sub-interfaces) or **frame-relay map**.
- The default is to use the Inverse ARP (IARP) to map the DLCI to the IP address of next-hop router. However, you can also configure static mapping using the **frame-relay map ipip-address dlci broadcast** command.
- There are two ways to associate one DLCI to point-to-point or multiple DLCIs to multipoint interfaces. The first involves using the **frame-relay interface-dlci dlci** sub-interface command. The second involves using the **frame-relay map ipip-address dlci broadcast** sub-interface command.

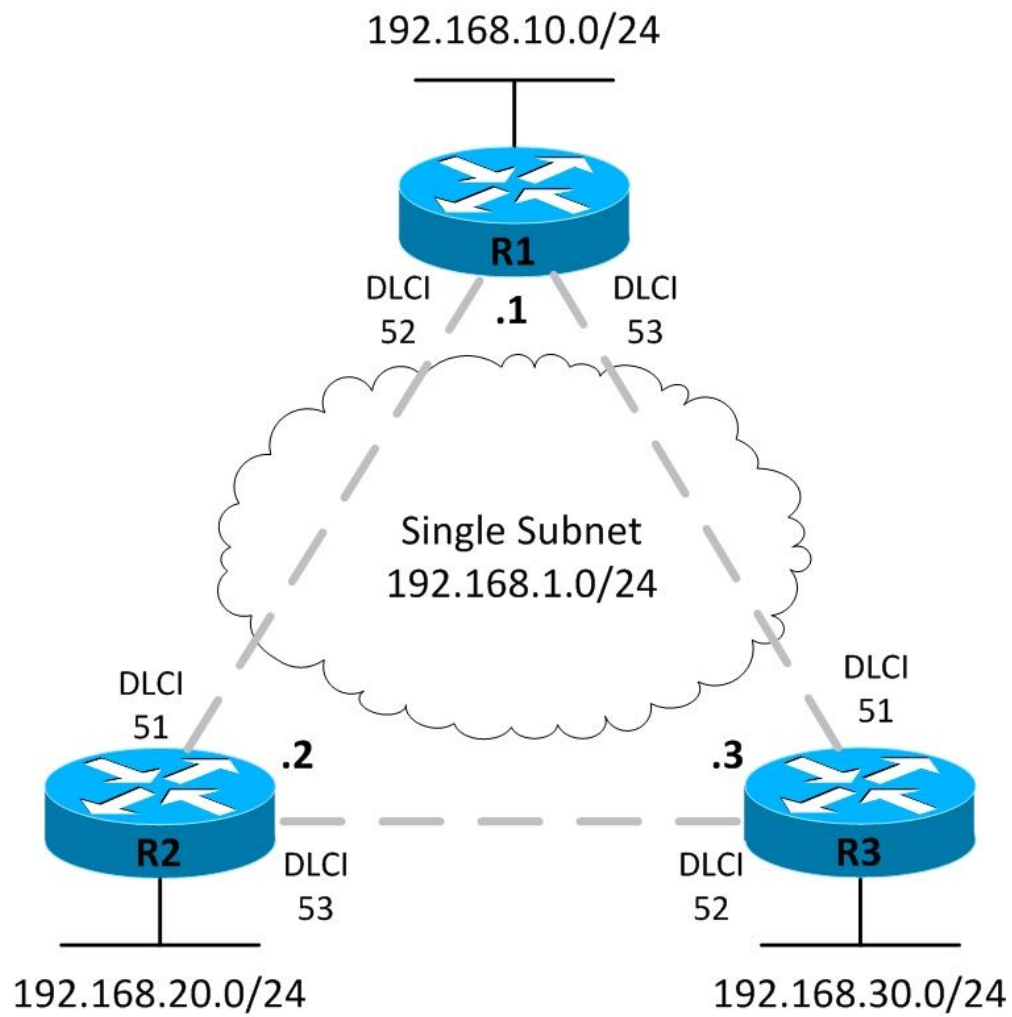
We are going to present three different Frame Relay configuration examples to see all those configuration steps in action. The examples correspond to the three Frame Relay scenarios we presented earlier in the chapter. We will also introduce you to several **show** commands that are useful to verify your configuration and troubleshoot if something is not working as expected.

### Configuration – Single Subnet for all Routers

The first option involves a single IP subnet for all routers/DTEs, with IP addresses configured on physical serial interfaces, as shown in Figure 12-17.

**Figure 12-17** Configuration – Single Subnet for all Routers





We will use a single class C private subnet 192.168.1.0/24 in this example. Table 12-5 should serve as a reference for all configuration in this section.

**Table 12-5** Configuration Table

Router	Interface / Type	DLCI	IP Address
R1	Serial 0/0 / physical	Learned via InARP	192.168.1.1/24
R2	Serial 0/0 / physical	Learned via InARP	192.168.1.2/24
R3	Serial 0/0 / physical	Learned via InARP	192.168.1.3/24

We are going to configure IP addresses on physical serial interfaces of all three routers. Also, we will not configure or map any DLCIs manually. We will rather rely on Inverse ARP, enabled by default on serial interfaces with Frame Relay encapsulation, for learning DLCIs. The router connected to the Frame Relay network learns DLCI information from the LMI status messages sent by the Frame Relay switch to the router.

The ultimate goal of a Frame Relay network is to enable hosts on a LAN communicate with hosts on remote LANs. We will use EIGRP to propagate routing information to achieve that goal. The configuration is pretty simple here and we are just enabling Frame Relay encapsulation using the **encapsulation frame-relay** command.

```
R1> enable
R1# configure terminal
R1(config)# interface Serial0/0
R1(config-if)# ip address 192.168.1.1 255.255.255.0
R1(config-if)# encapsulation frame-relay
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config-if)# interface FastEthernet1/0
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# router eigrp 100
R1(config-router)# network 192.168.1.0
R1(config-router)# network 192.168.10.0
R1(config-router)# end
R1#
```

The configuration for R2 is very similar.

```
R2> enable
R2# configure terminal
R2(config)# interface Serial0/0
R2(config-if)# ip address 192.168.1.2 255.255.255.0
R2(config-if)# encapsulation frame-relay
R2(config-if)# no shutdown
R2(config-if)# exit
R2(config-if)# interface FastEthernet1/0
R2(config-if)# ip address 192.168.20.1 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# exit
R2(config)# router eigrp 100
R2(config-router)# network 192.168.1.0
R2(config-router)# network 192.168.20.0
R2(config-router)# end
R2#
```

There are no surprises with the configuration of R3 either.

```
R3> enable
R3# configure terminal
R3(config)# interface Serial0/0
R3(config-if)# ip address 192.168.1.3 255.255.255.0
R3(config-if)# encapsulation frame-relay
R3(config-if)# no shutdown
R3(config-if)# exit
R3(config-if)# interface FastEthernet1/0
R3(config-if)# ip address 192.168.30.1 255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# exit
R3(config)# router eigrp 100
R3(config-router)# network 192.168.1.0
R3(config-router)# network 192.168.30.0
R3(config-router)# end
R3#
```

We are done with our Frame Relay configuration here, and it's time to verify if it works as expected. A good starting point for Frame Relay verification can be the **show frame-relay map command**.

```
R1#show frame-relay map
Serial0/0 (up): ip 192.168.1.2 dlci 52(0x34,0xC40), dynamic,
broadcast,, status defined, active
Serial0/0 (up): ip 192.168.1.3 dlci 53(0x35,0xC50), dynamic,
broadcast,, status defined, active
```

The above output is full of useful information. First, it tells you that two DLCIs are available on the interface Serial0/0 that correspond to two VCs. IP addresses 192.168.1.2 and 192.168.1.3 are dynamically mapped to DLCIs 52 and 53 respectively. The DLCIs are both learned from the Frame Relay switch through LMI Status messages sent from the switch to the router, and both are *active* which is the desired state.

Here is the output of **show frame-relay lmi** command executed on R1.

```
R1#show frame-relay lmi
```

```
LMI Statistics for interface Serial0/0 (Frame Relay DTE) LMI TYPE = CISCO
Invalid Unnumbered info 0      Invalid Prot Disc 0
Invalid dummy Call Ref 0      Invalid Msg Type 0
Invalid Status Message 0      Invalid Lock Shift 0
Invalid Information ID 0       Invalid Report IE Len 0
Invalid Report Request 0       Invalid Keep IE Len 0
Num Status Enq. Sent 9         Num Status msgs Rcvd 9
Num Update Status Rcvd 0       Num Status Timeouts 0
Last Full Status Req 00:00:24   Last Full Status Rcvd 00:00:24
```

The above output provides detailed LMI statistics for Frame Relay interfaces on the router, in this case only Serial0/0. The first thing to notice is that the LMI type is **cisco** which is expected as we did not explicitly configure it and the router defaulted to **cisco**. The number of status enquiry messages sent equals the number of status messages received from the Frame Relay switch. These numbers increment by one almost every 10 seconds under normal conditions. The last output line is interesting as it shows the time elapsed since the last *full* status was received. You may recall that regular status messages are received in response to status enquiry messages every 10 seconds. However there is a full status message sent by the Frame Relay switch every 60 seconds that include complete information about all DLCIs.

We will run a quick debug on R1 to see what LMI messages are being exchanged. We include a sample output of **debug frame-relay lmi** command for R1 here. You can see a **StEnq** (status enquiry) message sent out the interface Serial0/0 by the Frame Relay DTE/router. A **Status** message from Frame Relay DCE/switch arrives at interface Serial0/0 shortly after that.

## TUN MIN OO {BE-IT} Routing & Switching 200-120

```
R1#debug frame-relay lmi
Frame Relay LMI debugging is on
Displaying all Frame Relay LMI data
R1#
*Mar 1 00:00:40.295: Serial0/0(out): StEnq, myseq 3, yourseen 2, DTE up
*Mar 1 00:00:40.299: datagramstart = 0x7A00714, datagramsize = 13
*Mar 1 00:00:40.299: FR encap = 0xFCF10309
*Mar 1 00:00:40.303: 00 75 01 01 01 03 02 03 02
*Mar 1 00:00:40.307:
*Mar 1 00:00:40.327: Serial0/0(in): Status, myseq 3, pak size 13
*Mar 1 00:00:40.327: RT IE 1, length 1, type 1
*Mar 1 00:00:40.331: KA IE 3, length 2, yourseq 3, myseq 3
```

The **show frame-relay pvc** command can be used to view PVC status and some traffic statistics.

```
R1#show frame-relay pvc
```

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

Active	Inactive	Deleted	Static
Local	2	0	0
Switched	0	0	0
Unused	0	0	0

DLCI = 52, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

```
input pkts 17      output pkts 16      in bytes 1022
out bytes 986      dropped pkts 0      in pkts dropped 0
out pkts dropped 0  out bytes dropped 0
in FECN pkts 0     in BECN pkts 0     out FECN pkts 0
out BECN pkts 0     in DE pkts 0        out DE pkts 0
out bcast pkts 10   out bcast bytes 610
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:07:28, last time pvc status changed 00:07:28
```

DLCI = 53, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

```
input pkts 15      output pkts 16      in bytes 914
out bytes 958      dropped pkts 0      in pkts dropped 0
out pkts dropped 0  out bytes dropped 0
in FECN pkts 0     in BECN pkts 0     out FECN pkts 0
out BECN pkts 0     in DE pkts 0        out DE pkts 0
```

## TUN MIN OO {BE-IT} Routing & Switching 200-120

```
out bcast pkts 10      out bcast bytes 610
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 00:07:30, last time pvc status changed 00:07:30
```

We may examine the routing table of R1, which shows some EIGRP learned routes. It indicates that the PVCs are active and some routing information has been exchanged over those PVCs by EIGRP.

```
R1#show ip route
<Some output omitted for brevity.>
```

Gateway of last resort is not set

```
D  192.168.30.0/24 [90/2172416] via 192.168.1.3, 00:06:46, Serial0/0
C  192.168.10.0/24 is directly connected, FastEthernet1/0
D  192.168.20.0/24 [90/2172416] via 192.168.1.2, 00:07:39, Serial0/0
C  192.168.1.0/24 is directly connected, Serial0/0
```

The ultimate test is to verify end-to-end connectivity across all three VCs we have, which can be done by going to each of the three routers one by one and pinging the other two routers.

```
R1#ping 192.168.1.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/40/56 ms

We should also verify connectivity between the local-area networks (LANs) attached to routers.

```
R1#ping 192.168.20.1 source FastEthernet1/0
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.20.1, timeout is 2 seconds:

Packet sent with a source address of 192.168.10.1

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 36/40/44 ms

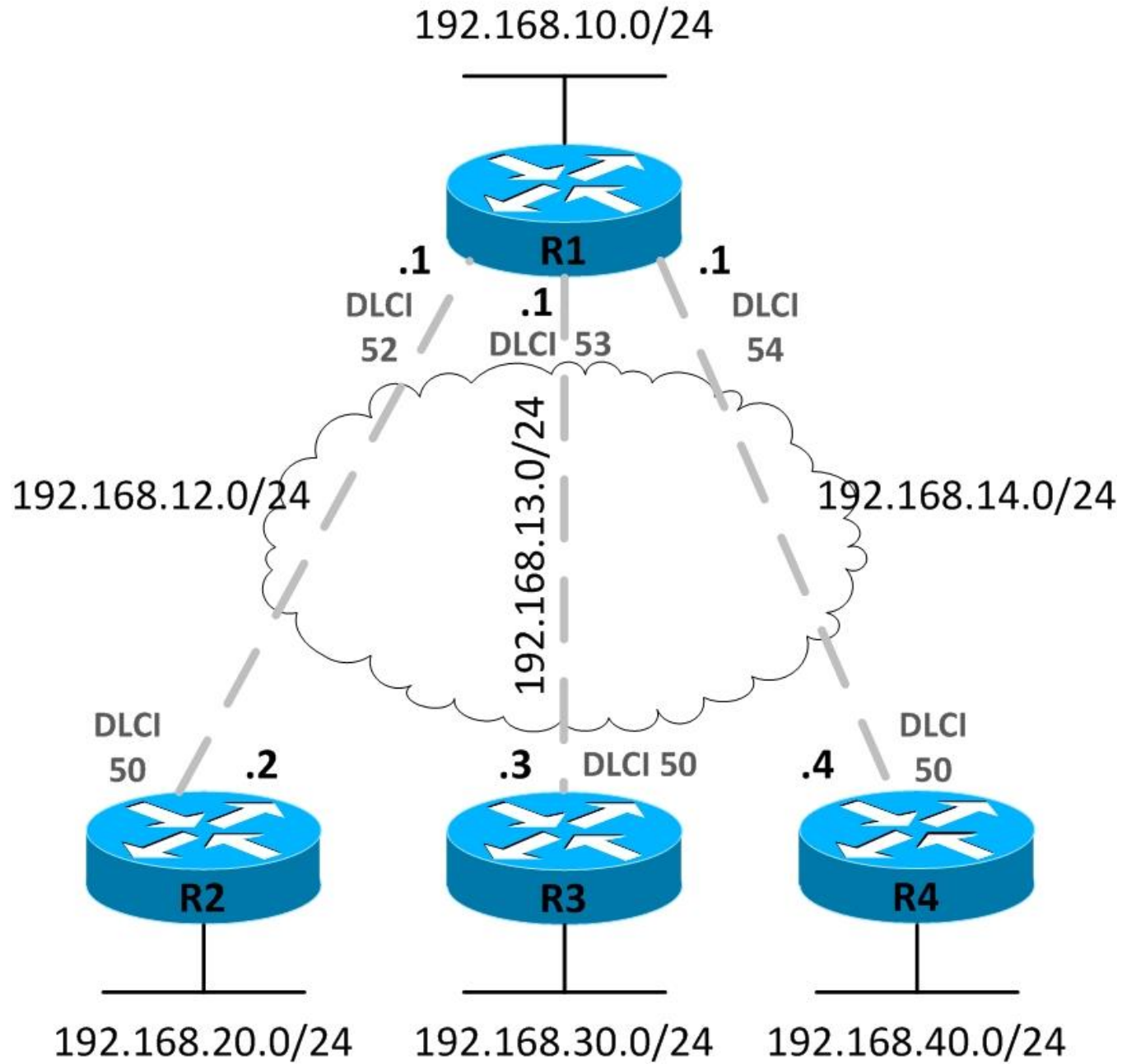


We are not including all the ping tests here for the sake of brevity, but we have achieved end-to-end reachability at this point.

### **Configuration – One Subnet per VC**

The second configuration example involves one subnet per virtual circuit, as shown in Figure 12-16. It is a special case of partial mesh topology also known as hub-and-spoke. We will also introduce you to some additional Frame Relay configuration options not seen in the first example above.

**Figure 12-18** Configuration – One Subnet per VC



We will use sub-interfaces in this example and manually assign DLCIs to sub-interfaces, per the following table.

**Table 12-6** Configuration Table

Router	Interface / Type	DLCI	IP Address
R1	Serial 0/0.2 / <b>point-to-point</b>	52	192.168.12.1/24
R1	Serial 0/0.3 / <b>point-to-point</b>	53	192.168.13.1/24
R1	Serial 0/0.4 / <b>point-to-point</b>	54	192.168.14.1/24
R2	Serial 0/0.1 / <b>point-to-point</b>	51	192.168.12.2/24
R3	Serial 0/0.1 / <b>point-to-point</b>	51	192.168.13.3/24
R4	Serial 0/0.1 / <b>point-to-point</b>	51	192.168.14.4/24

We are going to use point-to-point sub-interfaces, and DLCIs assigned manually to sub-interfaces. Even though the router can learn DLCIs through LMI messages, but those DLCI's will all be assigned to the physical interface by default, rather than point-to-point sub-interface. However Inverse ARP is still used to map remote IP addresses to DLCIs.

```
R1>enable
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface Serial0/0
R1(config-if)#encapsulation frame-relay
R1(config-if)#no shutdown
R1(config-if)#interface Serial0/0.2 point-to-point
R1(config-subif)#ip address 192.168.12.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 52
R1(config-fr-dlci)#interface Serial0/0.3 point-to-point
R1(config-subif)#ip address 192.168.13.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 53
R1(config-fr-dlci)#interface Serial0/0.4 point-to-point
R1(config-subif)#ip address 192.168.14.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 54
R1(config-fr-dlci)#interface FastEthernet1/0
R1(config-if)#ip address 192.168.10.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#router eigrp 100
R1(config-router)#network 192.168.12.0
R1(config-router)#network 192.168.13.0
R1(config-router)#network 192.168.14.0
R1(config-router)#network 192.168.10.0
```

## TUN MIN OO {BE-IT} Routing & Switching 200-120

```
R1(config-router)#end
R1#
```

R2 has a similar configuration.

```
R2>enable
R2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#
R2(config)#interface Serial0/0
R2(config-if)#encapsulation frame-relay
R2(config-if)#no shutdown
R2(config-if)#interface Serial0/0.1 point-to-point
R2(config-subif)#ip address 192.168.12.2 255.255.255.0
R2(config-subif)#frame-relay interface-dlci 51
R2(config-fr-dlci)#interface FastEthernet1/0
R2(config-if)#ip address 192.168.20.1 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#router eigrp 100
R2(config-router)#network 192.168.12.0
R2(config-router)#network 192.168.20.0
R2(config-router)#end
R2#
```

R3 has pretty much similar configuration as well.

```
R3>enable
R3#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#interface Serial0/0
R3(config-if)#encapsulation frame-relay
R3(config-if)#no shutdown
R3(config-if)#interface Serial0/0.1 point-to-point
R3(config-subif)#ip address 192.168.13.3 255.255.255.0
R3(config-subif)#frame-relay interface-dlci 51
R3(config-fr-dlci)#interface FastEthernet1/0
R3(config-if)#ip address 192.168.30.1 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#router eigrp 100
R3(config-router)#network 192.168.13.0
R3(config-router)#network 192.168.30.0
R3(config-router)#end
R3#
```

## TUN MIN OO {BE-IT} Routing & Switching 200-120

R4 too has a single PVC to R1 like R2 and R3. R1 happens to be the hub in this hub-and spoke topology. This topology is commonly used in real-world Frame Relay networks where a large number of remote offices are connected to the company headquarters.

```
R4>enable
R4#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#interface Serial0/0
R4(config-if)#encapsulation frame-relay
R4(config-if)#no shutdown
R4(config-if)#interface Serial0/0.1 point-to-point
R4(config-subif)#ip address 192.168.14.4 255.255.255.0
R4(config-subif)#frame-relay interface-dlci 51
R4(config-fr-dlci)#interface FastEthernet1/0
R4(config-if)#ip address 192.168.40.1 255.255.255.0
R4(config-if)#no shutdown
R4(config-if)#router eigrp 100
R4(config-router)#network 192.168.14.0
R4(config-router)#network 192.168.40.0
R4(config-router)#end
R4#
```

Let's verify that PVCs are established between R1 and the rest of routers by using the **show frame-relay map** command.

```
R1#show frame-relay map
Serial0/0.3 (up): point-to-point dlci, dlci 53(0x35,0xC50), broadcast
status defined, active
Serial0/0.2 (up): point-to-point dlci, dlci 52(0x34,0xC40), broadcast
status defined, active
Serial0/0.4 (up): point-to-point dlci, dlci 54(0x36,0xC60), broadcast
status defined, active
```

Let's examine the status of Frame Relay sub-interfaces on R1.

```
R1#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Serial0/0	unassigned	YES	NVRAM	up	up
Serial0/0.2	192.168.12.1	YES	NVRAM	up	up
Serial0/0.3	192.168.13.1	YES	NVRAM	up	up
Serial0/0.4	192.168.14.1	YES	NVRAM	up	up

Serial0/1	unassigned	YES	NVRAM	administratively down	down
Serial0/2	unassigned	YES	NVRAM	administratively down	down
Serial0/3	unassigned	YES	NVRAM	administratively down	down
FastEthernet1/0	192.168.10.1	YES	NVRAM	up	up

The PVCs are active and we have end-to-end connectivity at this point.

### Configuration – A Mix of Full and Partial Mesh

The third and last configuration example involves a mix of full and partial mesh, as shown in Figure 12-17. We are going to have both **point-to-point** and **multipoint** sub-interfaces. Multipoint interfaces are logical Frame Relay sub-interfaces but they can terminate more than one PVCs just like physical serial interfaces.

**Figure 12-19** Configuration – A Mix of Full and Partial Mesh

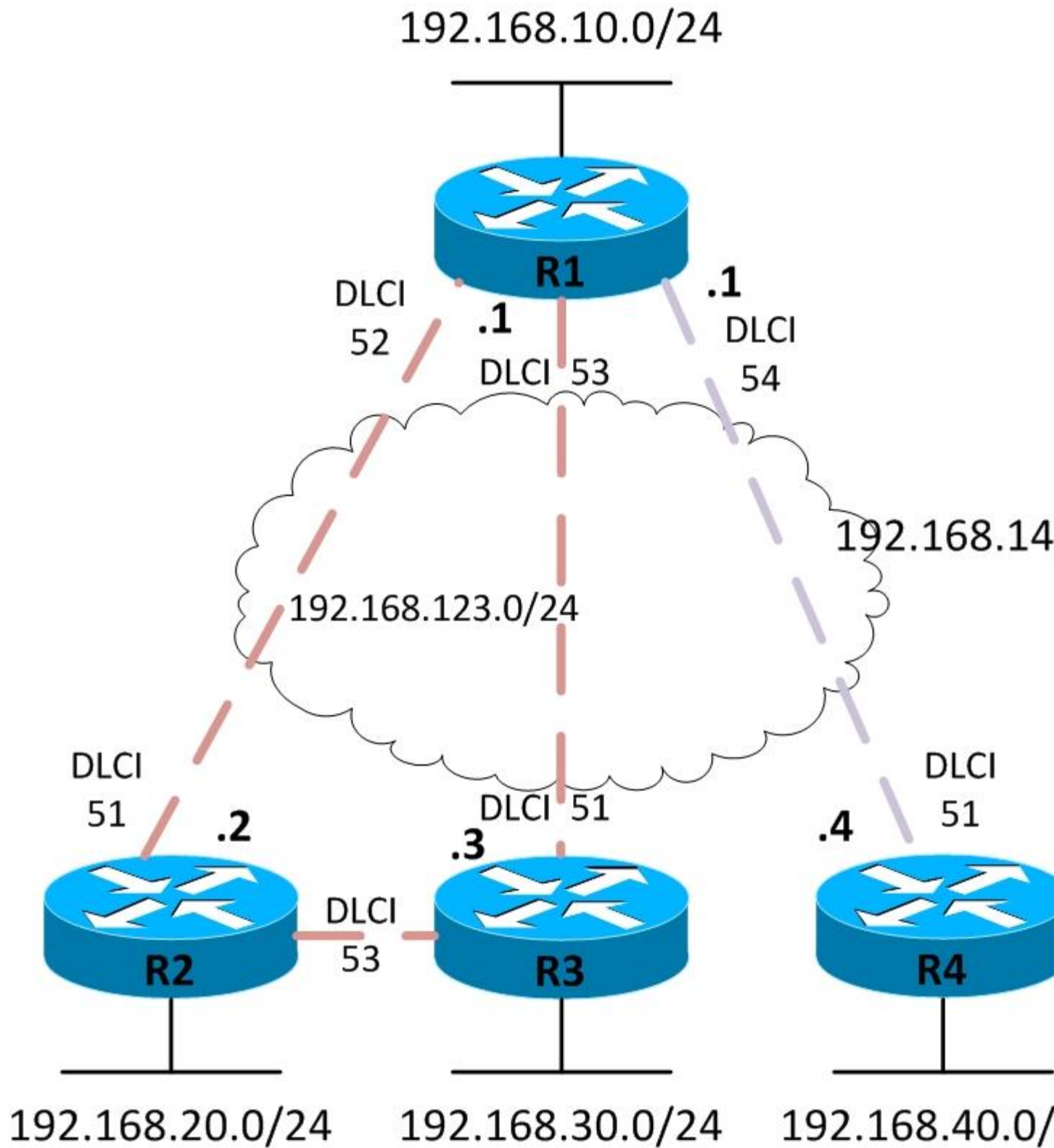


Table 12-7 Configuration Table

Router	Interface / Type	DLCI	IP Address
R1	Serial 0/0.123 / <b>multipoint</b>	52, 53	192.168.123.1/24
R1	Serial 0/0.4 / <b>point-to-point</b>	54	192.168.14.1/24
R2	Serial 0/0.123 / <b>multipoint</b>	51, 53	192.168.123.2/24
R3	Serial 0/0.123 / <b>multipoint</b>	51, 53	192.168.123.3/24
R4	Serial 0/0.1 / <b>point-to-point</b>	51	192.168.14.4/24

R1 has a multipoint Frame Relay sub-interface connected to subnet 192.168.123.0/24 while a point-to-point subinterface terminates the PVC to R4.

```

R1>enable
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface Serial0/0
R1(config-if)#encapsulation frame-relay
R1(config-if)#no shutdown
R1(config-if)#interface Serial0/0.4 point-to-point
R1(config-subif)#ip address 192.168.14.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 54
R1(config-fr-dlci)#interface Serial0/0.123 multipoint
R1(config-subif)#ip address 192.168.123.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 52
R1(config-fr-dlci)#frame-relay interface-dlci 53
R1(config-fr-dlci)#interface FastEthernet1/0
R1(config-if)#ip address 192.168.10.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#router eigrp 100
R1(config-router)#network 192.168.14.0
R1(config-router)#network 192.168.123.0
R1(config-router)#network 192.168.10.0
R1(config-router)#end
R1#

```

R1 has a multipoint Frame Relay sub-interfaces connected to the subnet 192.168.123.0/24 as well.

```

R2>enable
R2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#interface Serial0/0
R2(config-if)#encapsulation frame-relay
R2(config-if)#no shutdown

```



## TUN MIN OO {BE-IT} Routing & Switching 200-120

```
R2(config-if)#interface Serial0/0.123 multipoint
R2(config-subif)#ip address 192.168.123.2 255.255.255.0
R2(config-subif)#frame-relay interface-dlci 51
R2(config-fr-dlci)#frame-relay interface-dlci 53
R2(config-fr-dlci)#interface FastEthernet1/0
R2(config-if)#ip address 192.168.20.1 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#router eigrp 100
R2(config-router)#network 192.168.123.0
R2(config-router)#network
R2(config-router)#end
R2#
```

R3 also shares the subnet 192.168.123.0/24 via its Frame Relay multipoint sub-interface that terminates two PVCs.

```
R3>enable
R3#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#interface Serial0/0
R3(config-if)#encapsulation frame-relay
R3(config-if)#no shutdown
R3(config-if)#interface Serial0/0.123 multipoint
R3(config-subif)#ip address 192.168.123.3 255.255.255.0
R3(config-subif)#frame-relay interface-dlci 51
R3(config-fr-dlci)#frame-relay interface-dlci 52
R3(config-fr-dlci)#interface FastEthernet1/0
R3(config-if)#ip address 192.168.30.1 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#router eigrp 100
R3(config-router)#network 192.168.123.0
R3(config-router)#network 192.168.30.0
R3(config-router)#end
R3#
```

R4 has a point-to-point sub-interface only terminating a PVC to R1.

```
R4>enable
R4#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#interface Serial0/0
R4(config-if)#encapsulation frame-relay
R4(config-if)#no shutdown
```

## TUN MIN OO {BE-IT} Routing & Switching 200-120

```
R4(config-if)#interface Serial0/0.1 point-to-point
R4(config-subif)#ip address 192.168.14.4 255.255.255.0
R4(config-subif)#frame-relay interface-dlci 51
R4(config-fr-dlci)#interface FastEthernet1/0
R4(config-if)#ip address 192.168.40.1 255.255.255.0
R4(config-if)#no shutdown
R4(config-if)#router eigrp 100
R4(config-router)#network 192.168.14.0
R4(config-router)#network 192.168.40.0
R4(config-router)#end
R4#
```

It's time to view the Frame Relay DLCI to IP address mappings learned via InARP, using **show frame-relay map** command on R1.

```
R1#show frame-relay map
Serial0/0.123 (up): ip 192.168.123.2 dlci 52(0x34,0xC40), dynamic,
broadcast,, status defined, active
Serial0/0.123 (up): ip 192.168.123.3 dlci 53(0x35,0xC50), dynamic,
broadcast,, status defined, active
Serial0/0.4 (up): point-to-point dlci, dlci 54(0x36,0xC60), broadcast
status defined, active
```

We will use the **show frame-relay pvc** command on R4, to examine the status of PVC.

```
R4#show frame-relay pvc
```

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

Active	Inactive	Deleted	Static
Local	1	0	0
Switched	0	0	0
Unused	0	0	0

DLCI = 51, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

input pkts 17	output pkts 23	in bytes 1192
out bytes 2970	dropped pkts 0	in pkts dropped 0
out pkts dropped 0	out bytes dropped 0	
in FECN pkts 0	in BECN pkts 0	out FECN pkts 0
out BECN pkts 0	in DE pkts 0	out DE pkts 0

## TUN MIN OO {BE-IT} Routing & Switching 200-120

out bcast pkts 17      out bcast bytes 2630  
5 minute input rate 0 bits/sec, 0 packets/sec  
5 minute output rate 0 bits/sec, 0 packets/sec  
pvc create time 00:00:59, last time pvc status changed 00:00:39

## 11-14 Other WAN Technologies

We have offered in-depth coverage of the following WAN technologies so far in this chapter: HDLC, PPP, and Frame Relay.



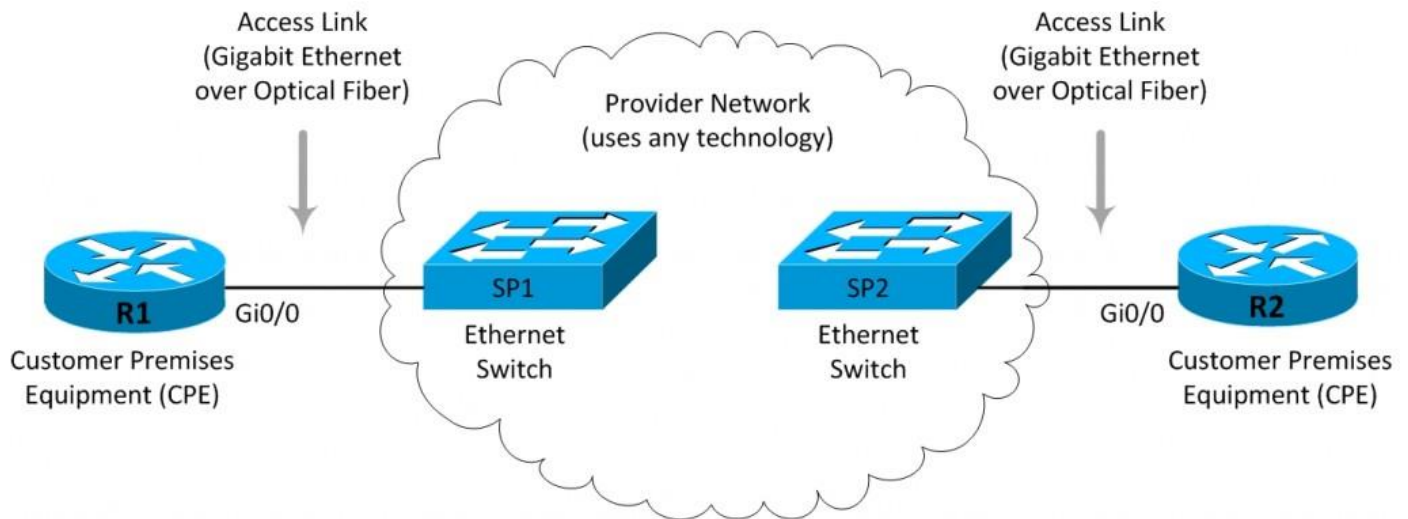
**Key Concept** The most widely used data link protocols used on serial WAN links are HDLC, PPP, Frame Relay.

In this section, we will briefly introduce you to a handful of other WAN protocols.

### Ethernet WANs

Ethernet began life as a LAN technology and remained so for quite a while because distance limitations made it difficult to create longer links. However, Ethernet standards kept improving with time, both in speed and distance, especially for optical fiber media. The result is that service providers now can and do offer WAN services that employ Ethernet both on the edge for customer access links and in the core of the provider network.

**Figure 12-20** Ethernet WAN Service



Different kinds of Ethernet WAN services are commercially available with many different names such as Wide Area Ethernet, Ethernet over MPLS (EoMPLS), Metropolitan Ethernet (MetroE), and Virtual Private LAN Service (VPLS). In fact the provider can use any technology inside its network to create an Ethernet WAN service for its customers. Ethernet WAN services usually offer 100 Mbps or 1 Gbps speeds to customers.

## **Multi-Protocol Label Switching (MPLS)**

Multiprotocol Label Switching (MPLS) technology is used by service providers to offer many types of WAN services. We will mention one of those WAN services called MPLS VPN that happens to be very popular with enterprise customers. MPLS VPN has a familiar service model, with customer sites connecting to the provider's network cloud and the cloud moving data between customer sites connected to the cloud as required. The service provider also promises to keep data from different customers separate as it passes through its network.

MPLS VPNs have many differences from other WAN services, but the most significant difference is that they are aware of IP packets from customers. They do not just promise to deliver bits like leased lines or data link frames like Frame Relay and Ethernet WAN. MPLS network is more like an IP network, routing IP packets between customers sites. Due to this IP awareness of the MPLS network, service providers are offering many interesting services to customers.

## **Digital Subscriber Line (DSL)**

Digital Subscriber Line (DSL) has enabled much faster Internet access speeds to both homes and businesses as compared with dial-up and ISDN technologies that DSL has almost completely replaced now.

One limitation of DSL is that it only works at certain distances from the central office (CO) to the home and as cable distance increases it suffers speed degrades. So if the site where you want to have a DSL connection happens to be far from the nearest CO, the quality of the service may become poor or the service may not be available at all. Though this is usually not a concern in urban areas, yet you may occasionally see this problem.

## **PPP over Ethernet (PPPoE)**

PPP over Ethernet (PPPoE) is one technology overlaid on top of another. You know that PPP is a data link protocol used on serial interfaces to create point-to-point links over leased lines. PPP is also used on those links that are created from a user to an ISP with dial-up modems. Some features of PPP are very useful for ISPs. First, PPP supports a way to assign IP addresses to the other end of the PPP link. PPP also supports CHAP for authentication which allows ISPs to check their accounting records to see if the customer's bill was paid before granting Internet access.

DSL came after dial-up and ISDN that both used PPP, so ISPs still wanted their PPP with DSL. The customer however mostly used an Ethernet link between the customer PC or the router and the DSL modem. That Ethernet link only supported Ethernet data link protocols and not PPP. ISPs demanded a way to create the equivalent of a PPP connection between the customer router and the ISP router over the various technologies used on DSL connections.

PPP over Ethernet (PPPoE) was created to allow the sending of PPP frames encapsulated inside Ethernet frames. PPPoE essentially creates a tunnel between customer router and the ISP router.

PPP was originally meant for point-to-point links and there is not a single point-to-point link between the two routers here. With PPPoE and its associated protocols, the router logically creates a tunnel and then creates and sends PPP frames over that tunnel as if the tunnel were a point-to-point link between the routers.

### Summary

In this chapter, you learned about the following WAN technologies: High-Level Data Link Control (HDLC), Point-to-Point Protocol (PPP), and Frame Relay.

You learned that HDLC is a basic protocol for point-to-point serial links but if all that you need is to connect two routers over a leased line, HDLC is just fine and it's enabled by default on serial interfaces of Cisco routers. If you need more features than HDLC offers or if you are using two routers from different manufacturers, you should use PPP rather than Cisco-proprietary HDLC.

You were introduced to several PPP concepts including the role of LCP and different NCPs, one for each Layer 3 protocol encapsulated by PPP. You also learned about two types of authentication that can be used with PPP: Password Authentication Protocol (PAP) and Channel Handshake Authentication Protocol (CHAP).

We talked about Frame Relay in detail covering different encapsulation methods, addressing, LMI options, Frame Relay maps, and virtual circuits. We also learned in-depth how to configure and verify Frame Relay.

## ***Chapter 12 – Virtual Private Networks***

### **Chapter 12 – Virtual Private Networks**

- [12-1 VPN Concepts](#)
- [12-2 Types of VPN](#)
- [12-3 Encryption](#)
- [12-4 IPsec VPNs](#)
- [12-5 SSL VPNs & Tunneling Protocols](#)
- [12-6 GRE Tunnels](#)
- [12-7 VPN Summary](#)

## *12-1 VPN Concepts*

A company wanting to connect two (or more) of its sites can choose from several different types of WAN services: leased lines, Frame Relay, or more likely Multiprotocol Label Switching (MPLS) today. All these services are typically expensive. However, another much cheaper option exists for connecting company sites to each other. Each site can simply be connected to the Internet using a broadband Internet access technology like digital subscriber line (DSL), cable, WiMAX, or even 3G/4G. Different sites then can send data to each other using the public Internet as a wide area network (WAN).

There is one problem with using Internet as a WAN though. The Internet is not as secure as other WAN options. The vulnerability of the Internet is, to a great extent, due to the fact that it is a public network. Just anyone with a computer can access the Internet and possibly attack any other computer. Other WAN options mentioned here are relatively secure. For example, in order to steal data flowing over a leased line, the attacker has to physically tap into the line with specialized equipment or be present at the telco central office. These actions are punishable by law and not easy for just anyone.

The possibility to use the Internet as a WAN is quite tempting despite the security concerns. Virtual private network (VPN) technology provides answers to the security questions associated with using the Internet as a private WAN service. In this chapter, we introduce you to the basic concepts and terminology related to VPNs. We then discuss details of two main types of VPNs: IP Security (IPsec) and Secure Sockets Layer (SSL).

VPNs have several advantages over other WAN technologies, some of which are summarized here:

- **Cost:** Internet VPN solutions can be much cheaper than alternate private WAN options available today.
- **Security:** Modern VPN solutions can be as secure as private WAN options and are being used even by organizations with the most stringent security requirements such as credit card companies.
- **Scalability:** Internet VPN solutions can be scaled quickly and cost-effectively to a large number of sites. Each location can choose from multiple options of Internet connectivity.

## **VPN Concepts**

A virtual private network (VPN) is used to transport data from a private network to another private network over a public network, such as the Internet, using encryption to keep the data



confidential. In other words, a VPN is an encrypted connection between private networks over a public network, most often the Internet. VPNs provide the following services:

- **Confidentiality:** VPNs prevent anyone in the middle of the Internet from being able to read the data. The Internet is inherently insecure as data typically crosses networks and devices under different administrative controls. Even if someone is able to intercept data at some point in the network they won't be able to interpret it due to encryption.
- **Integrity:** VPNs ensure that data was not modified in any way as it traversed the Internet.
- **Authentication:** VPNs use authentication to verify that the device at the other end of VPN is a legitimate device and not an attacker impersonating a legitimate device.
- **Anti-Replay:** VPNs ensure that hackers are not able to make changes to packets that flow from source to destination. .

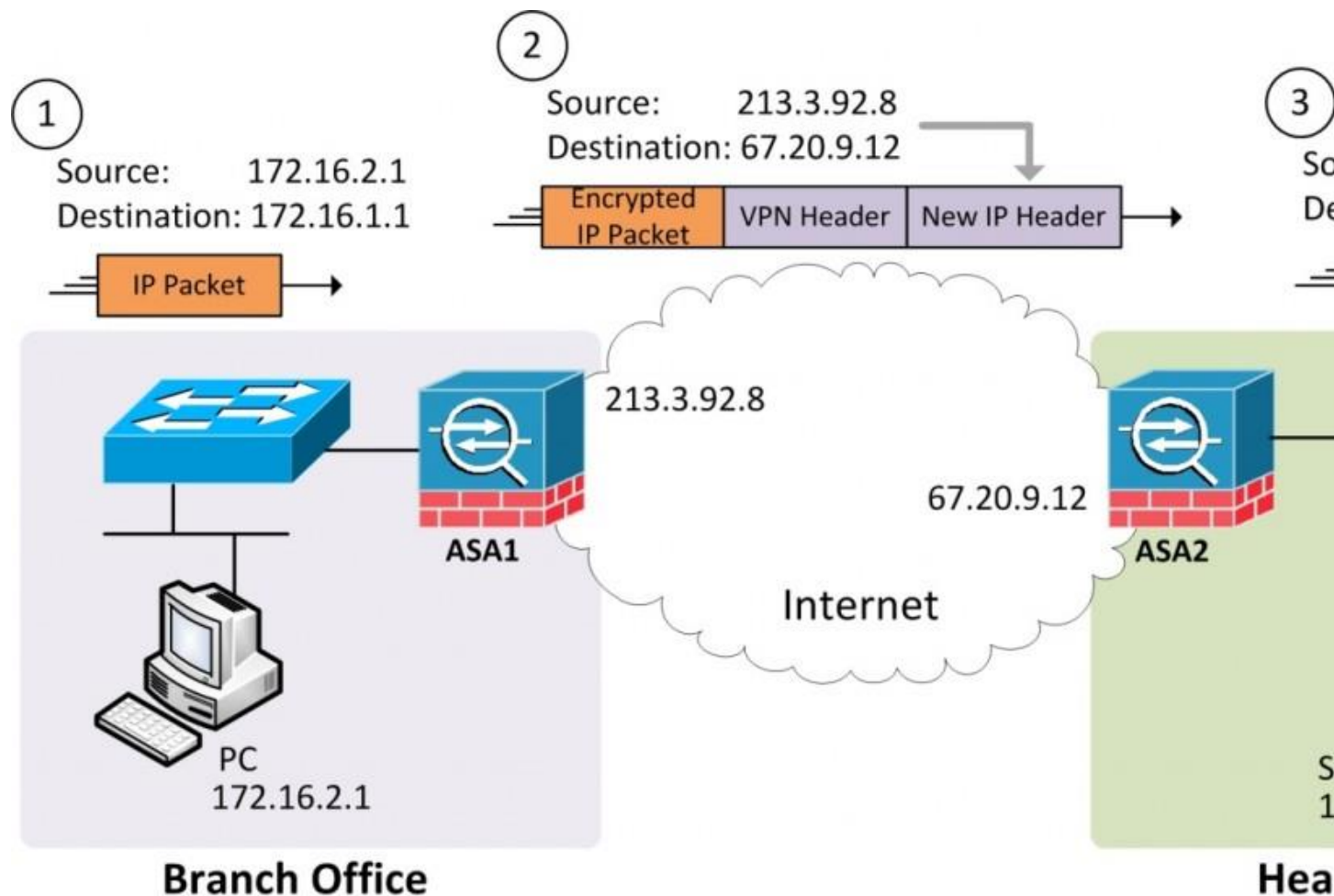


**Key Concept:** VPNs offer confidentiality, integrity, authentication, and anti-replay protection for user data.

A VPN is essentially a secure channel, often called a tunnel, between two devices or end points near the edge of the Internet. The VPN end points encrypt the whole of original IP packet, meaning the contents of the original packet cannot be understood by someone who even manages to see a copy of the packet as it traverses the network. The VPN end points also append headers to the original encrypted packet. The additional headers include fields that allow VPN devices to perform all their functions.

The graphic below and the explanation that follows should help you grasp basic VPN operation.

**Figure 12-1** VPN Concepts for a Site-to-Site VPN



The steps in the above graphic are explained here:

1. A PC in the branch office sends a packet to a server in the headquarters, just as it would without a VPN.
2. Cisco Adaptive Security Appliance (ASA) at the branch office, that is ASA1, encrypts the original packet, adds a VPN header, and adds a new IP header with public IP addresses.
3. ASA2 at the headquarters receives the packet, authenticates the identity of the sender, confirms that the packet has not been changed in transit, and then decrypts the original packet.
4. The server receives the decrypted packet.

Figure 12-1 shows Cisco Adaptive Security Appliance (ASA) performing VPN functions. However, several other hardware and software products are available for building VPNs. Some VPN products offered by Cisco are mentioned here.

- **Cisco Router:** All Cisco routers that run Cisco IOS software can support IPsec VPNs. The only requirement is that you should use a Cisco IOS image with appropriate feature

set. Examples of VPN-enabled routers include the Cisco 1800, Cisco 2800, Cisco 1900, and Cisco 2900 series.

- **Cisco Adaptive Security Appliance (ASA):** The Cisco ASA is a versatile appliance that combines several security functions including firewall and VPN capabilities in a single piece of hardware. All ASA models support IPsec VPN provided you meet the licensing requirements to enable the VPN feature.
- **Cisco VPN Clients:** Cisco offers both hardware and software VPN clients. *Cisco AnyConnect Secure Mobility Client* is a software VPN client that runs on laptops as well as smartphones and tablets.

## **12-2 Types of VPN**

There are two basic types of VPNs:

- Site-to-Site VPNs
- Remote Access VPNs

### **Site-to-Site VPNs**

A site-to-site VPN connects an entire network to another network. For example, they can connect a branch office network to the network at company headquarters, a VPN scenario also presented in Figure 13-1 earlier in this chapter. In the past, a private leased line or Frame Relay connection was required to connect sites. However, with easily available and cost-effective high-bandwidth Internet connections today, site-to-site VPNs can replace leased lines and Frame Relay.

Site-to-site VPNs are sometimes further classified as intranet and extranet VPNs. If a remote site of a company connects to the corporate headquarters of the same company, it is called an intranet VPN. When a company connects to a supplier, it is called an extranet VPN. From a technical standpoint these two types are the same though the distinction is important for your CCNA exam.

In site-to-site VPNs, a VPN gateway is installed at each site that performs encryption, decryption, and other services on behalf of all hosts on the local network. There is a variety of devices that can be configured to act as a VPN gateway such as a router, firewall, VPN concentrator, or another security appliance by Cisco or another manufacturer. The VPN gateway is responsible for encrypting and encapsulating the aggregate of all traffic going out from hosts on the local network and sending it through a VPN tunnel over the Internet to a peer VPN gateway at the target site. When the peer VPN gateway receives the traffic, it decapsulates and decrypts the content and forwards the packet toward the target host on its local inside private network.

### **Remote Access VPNs**

Remote access VPNs are analogous to circuit switched technologies such as dial-up connections and Integrated Services Digital Network (ISDN). Remote access VPNs fulfill the needs of mobile users and telecommuters working from home. Remote access VPNs connect individual hosts, rather than whole networks in the case of site-to-site VPNs, who must access their company network securely over the public Internet.

In a remote access VPN, the VPN client software is installed on each host. Whenever, the host has traffic to send, the VPN client software encapsulates and encrypts that traffic before sending it out the Internet to the VPN gateway at the entrance of the target network. The VPN gateway at the target network treats this traffic the same way as it does for site-to-site VPNs.



**Key Concept :** VPNs are classified as site-to-site VPNs that connect all the computers at two sites and remote access VPNs that connect individual users to a company network over the Internet. Site-to-site VPNs can be either intranet or extranet VPNs depending on if the two sites belong to the same or different partnering organizations respectively.



## **12-3 Encryption**

Encryption is the fundamental mechanism used to secure communications and is at the heart of any type of VPN implementation. Encryption obscures information to make it unreadable to unauthorized recipients. It provides a means to secure communications over an insecure medium such as the Internet. Let's now establish the definitions of some basic terms:

- **Plaintext:** The original data before encryption is known as plaintext.
- **Ciphertext:** The data after encryption is called ciphertext.
- **Hash:** A hash, or hash value, is a binary number generated from original data by applying a mathematical formula. Hash is a value calculated from the original data to uniquely identify the data.
- **Encryption:** It is the process that transforms plaintext into ciphertext. Encryption involves the use of an algorithmic process that uses a secret key (binary string) to transform plain data into a secret code.
- **Decryption:** It is the reverse process of encryption that is used to convert encrypted data back into its original form.

### **Cryptography Algorithms**

In general, there are three types of cryptography algorithms:

- **Symmetric Key Cryptography:** It involves a single key that is used for both encryption and decryption.
- **Asymmetric Key Cryptography:** It uses a pair of two different keys, one used for encryption and the other for decryption.
- **Hash Function:** A hash function is a one-way mathematical function that is used to produce a unique hash value from original data. The hash function is not reversible which means that the original data cannot be reconstituted from the hash value even with the knowledge of the hash function. The hash value is usually appended to the original message as the unique identifier of the message like a fingerprint.

In Figure 12-2, we present a very simple encryption algorithm known as the Caesar cipher. This method is named after Julius Caesar, who used it to encrypt his private correspondence. Each alphabet is shifted right or left by a fixed number of positions. The number of positions and the direction of shift must be known to both the sender and receiver in order to encrypt and decrypt the message.

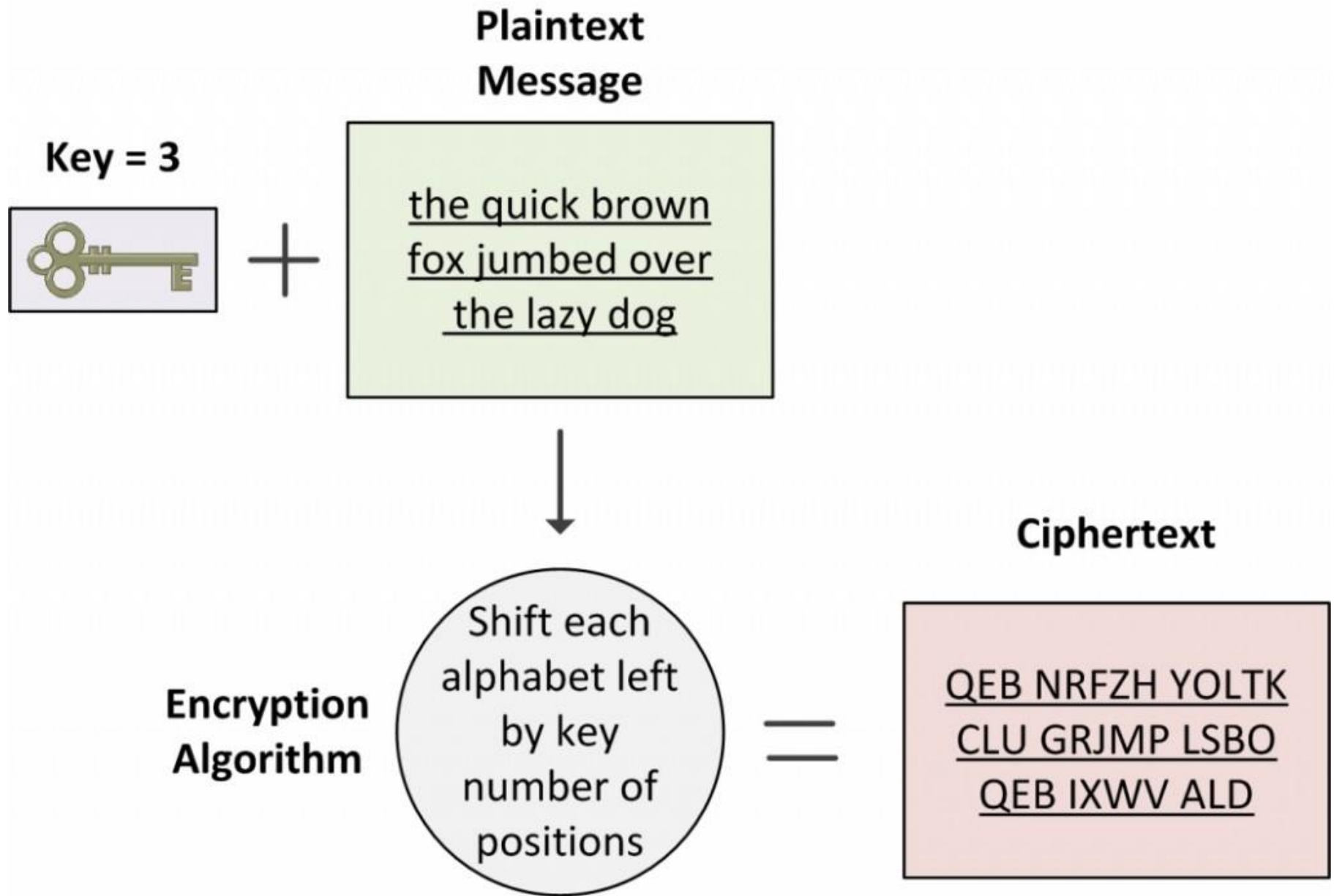
Caesar cipher with a left shift of three positions looks like this:

**Plaintext:**    ABCD EFGH IJKL MNOP QRST UVWX YZ  
**Ciphertext:** XYZA BCDE FGHI JKLM NOPQ RSTU VW

**Figure 12-2** Encryption Process







Please keep in mind that today's encryption algorithms are way more complex than the Caesar cipher and involve complicated mathematical computations that can be performed only by computers. However, the basic principle of encryption is still the same.

Symmetric key cryptography does not require a lot of computational power and therefore is much faster. It is well suited for encrypting large amounts of data such as data transfers over VPN connections. It can also run on network devices even without dedicated cryptography hardware due to being less computationally intensive. It should not be a surprise that symmetric key cryptography is employed by the most popular cryptographic algorithms today namely DES, 3DES, and AES.

- **Data Encryption Standard (DES):** DES is an old and common cryptographic algorithm. It uses a 56-bit key to encrypt 64-bit data blocks. DES is no longer considered very secure and is not recommended any more. The weakness of the protocol is primarily due to the very short key size of 56-bits.
- **Triple DES (3DES):** 3DES is an enhancement of DES that employs up to three 56-bit keys (168-bits). It runs three passes of the encryption and decryption process over the same block of data. DES was considered insecure due to its small key length of 56-bits. 3DES was derived from DES mainly to increase the length of key to 168-bits (three times the 56-bit key for DES) without switching over to an entirely new algorithm. 3DES also encrypts 64-bit data blocks just like DES, though it uses a 168-bit key. 3DES is the recommended replacement protocol to use in all DES implementations.
- **Advanced Encryption Standard (AES):** Advanced Encryption Standard (AES), also known as Rijndael, is one of the most common cryptography algorithms today. The AES is more flexible than both DES and 3DES as it uses a variable data block length as well as key length. It can use any combination of key lengths of 128, 192, or 256 bits and data block lengths of 128, 192, or 256 bits. AES is gradually replacing the predecessor DES and 3DES standards.

The following table provides a comparison of the three encryption algorithms at a glance.

**Table 12-1** Encryption Algorithms for VPNs

Algorithm	Key Length (bits)	Block Length (bits)	Security
DES	56	64	Insecure
3DES	168 (3 times 56)	64	Relatively secure
AES	128, 192, or 256	128, 192, or 256	Strong

Asymmetric key cryptography, also known as public-key cryptography, uses a two-key pair: one key is used to encrypt plaintext while the other key is used to decrypt the ciphertext. Each end user has its own pair of public and private keys. The public key of each end user is publicly available via a key management system. The private key is known only to the end user and is never exchanged or revealed to anyone other than the end user.

## 12-4 IPsec VPNs

IPsec derives its name from the title of RFC 4301, that is, Security Architecture for the Internet Protocol. IPsec is a set of security protocols that work together to ensure security of IP traffic as it traverses the Internet.

IPsec can be used to secure IP traffic between:

- Two hosts
- Two security gateways (usually routers or firewalls)
- A host and a security gateway

IPsec not only provides encryption at the network/IP layer but also defines a new set of headers that are added to the encrypted IP packet. IPsec is flexible being a framework of open standards, and describes the *messaging* to secure communications, but relies on existing algorithms.

IPsec uses the concept of a security association (SA) to define a set of security parameters used for various VPN functions. SAs are used by AH and ESP as well as by the IKE protocol. SAs are created as a result of an IPsec VPN connection establishment between two hosts or two gateways. SAs are uni-directional in nature and there will be two SAs in place with each secure connection, one for each direction.

There are three main protocols in the IPsec framework.

### **Internet Key Exchange (IKE)**

Internet Key Exchange (IKE) is a combination of Internet Security Association and Key Management Protocol (ISAKMP), Oakley, and SKEME protocols. The names IKE and ISAKMP are sometimes used interchangeably in IPsec discussions though we prefer to use IKE in this chapter. IKE establishes authenticated keys and also negotiates security associations (SAs) that are then used by ESP and AH protocols. IKE uses UDP port 500.

IKE is a two-phase protocol: IKE phase 1 verifies the identity of the remote peer or in other words authenticates the remote peer. The two peers then establish an authenticated secure channel to communicate further. IKE offers two primary methods of authenticating a remote peer:

- **Preshared Keys:** It is the most common method that uses manually configured secret keys on both peers. It is easy to deploy but is not scalable and very secure.
- **Public Key Signature:** It uses the Public Key Infrastructure (PKI) and is the most secure method.

At the end of phase 1 negotiation, an ISAKMP/IKE SA (phase 1 SA) is established. Phase 2 negotiations then take place over the secure channel established in phase 1.

IKE phase 2 negotiates SAs that are used to protect actual user data. At the end of phase 2 negotiations, two unidirectional IPsec SAs (phase 2 SAs) are established for user data. One SA is used for sending encrypted data and the other is used for receiving encrypted data.

### Authentication Header (AH)

Authentication Header (AH) provides data integrity and authentication for IP packets passed between two systems. It can be used when confidentiality is not required. It is used to verify that a message that has been passed from router A to router B has not been modified during transit. AH does not provide confidentiality and does not use encryption. All messages are sent in clear text, if the AH protocol is used alone which offers only weak security. However, AH is used in combination with other protocols like ESP to offer more robust security features.

### Encapsulating Security Payload (ESP)

Encapsulating Security Protocol (ESP) is a member of the IPsec protocol suite. It is an IP based protocol that uses IP port number 50 for communication between IPsec peers. It can provide authentication, integrity, confidentiality, and anti-replay protection of data. IP packet encryption not only hides the contents of the packet but also conceals the identities of the real source and destination found in the IP header in the form of source and destination IP addresses. ESP provides authentication for the encrypted IP packet and the ESP header. Authentication ensures data originated at a trusted source and was not modified during transit.

### IPsec Modes

IPsec has two modes of operation:

- **Tunnel Mode:** Tunnel mode secures data in site-to-site or network-to-network scenarios. In tunnel mode, the device performing VPN functions, such as a router or security appliance, does that on behalf of other users. In tunnel mode, the entire IP packet including the original IP header and the payload is encrypted and a new IP header is appended.
- **Transport Mode:** Transport mode secures data in host-to-host or end-to-end scenarios. In transport mode each user performs VPN functions on its own. In transport mode, IPsec protects the payload of the original IP packet but excludes the IP header. The transport mode, unlike the tunnel mode, preserves the original IP header and inserts the IPsec header between the original IP header and payload.

Both tunnel mode and transport mode can make use of ESP and AH protocols.

Cisco IOS defines bundles of encryption algorithms called transform sets that are used together to secure VPN traffic. IPsec transform sets define encapsulation (ESP or AH), encryption (3DES or AES-128), authentication/integrity algorithm (MD5 or SHA-1), and the IPsec mode (transport or tunnel). You have the option to create your own custom transform sets though Cisco IOS also provides some defaults.

**Table 12-2** Default IPsec Transform Sets in Cisco IOS

<b>Priority</b>	<b>Encapsulation</b>	<b>Encryption Algorithm</b>	<b>Hash Algorithm</b>
Higher	ESP	3DES	SHA-1
Lower	ESP	AES-128	SHA-1